

UNIVERSIDAD NACIONAL DE PIURA

FACULTAD DE CIENCIAS

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES



PROYECTO DE INVESTIGACION PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO Y DE TELECOMUNICACIONES

**“DISEÑO DE UN PROGRAMADOR LÓGICO PROGRAMABLE USANDO
MICROCONTROLADOR ATMEGA Y LENGUAJE LADDER PARA
APLICACIONES DE LABORATORIO”**

AUTORES:

Br. Ismael Sosa Yarlequé

ASESOR:

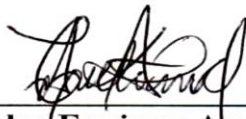
Ing. Eduardo Omar Ávila Regalado

Línea de Investigación: Información, Electrónica y Telecomunicaciones

Sub Línea de Investigación: Sistemas Digitales

PIURA NOVIEMBRE 2018

Señores jurados:



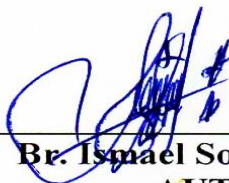
Dr. Carlos Enrique Arellano Ramírez
PRESIDENTE DE JURADO DE TESIS



Ing. Edwin Arnaldo Ocas Infante
SECRETARIO DE JURADO DE TESIS



Mg. Franklin Barra Zapata
VOCAL DE JURADO DE TESIS



Br. Ismael Sosa Yarlequé
AUTOR



Ing. Eduardo Omar Ávila Regalado
ASESOR

DECLARACIÓN JURADA DE AUTENTICIDAD DE LA TESIS

Yo: Ismael Sosa Yarlequé con DNI N° 44069839, Bachiller de Escuela Profesional de Ingeniería Electrónica y Telecomunicaciones, de la Facultad de Ciencias y domiciliado en Caserío Villa La Legua 497, Catacaos, Piura, Celular: 991026782.

Email: syi27@hotmail.com

DECLARO BAJO JURAMENTO: que la tesis que presento es auténtica e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art. N° 411, del código Penal concordante con el Art. 32° de la Ley N° 27444, y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo cual firmo la presente.

Piura, Septiembre del 2018


Ismael Sosa Yarlequé

Artículo 411.- El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 4. Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales –RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD

CARTA DE COMPROMISO DEL ASESOR

“AÑO DEL DIALOGO Y LA RECONCILIACION NACIONAL”

Quien suscribe, Ing. Eduardo Omar Ávila Regalado, con Documento de Identidad Nacional N°18077678, mediante la presente manifiesto que he leído y revisado de manera detallada el proyecto de investigación titulado: “DISEÑO DE UN PROGRAMADOR LÓGICO PROGRAMABLE USANDO MICROCONTROLADOR ATMEGA Y LENGUAJE LADDER PARA APLICACIONES DE LABORATORIO”, presentado por el tesista Bach. Ismael Sosa Yarlequé, identificado con Documento de Identidad Nacional N° 44069839, egresado de la carrera profesional de Ingeniería Electrónica y Telecomunicaciones, para optar el título profesional de Ingeniero Electrónico y Telecomunicaciones

En mi condición de asesor, considero que el mencionado proyecto, cumple con los requisitos exigidos por la comunidad científica, lo establecido en el Reglamento de Tesis para optar el título profesional en la UNP y amerita su ejecución, por lo que me comprometo a asesorar hasta la sustentación y publicación, si fuera el caso.

Piura-Perú, Septiembre, 2018.

Ing. Eduardo Omar Ávila Regalado

Asesor

DEDICATORIA

A lo largo de estos años he tenido la alegría de compartir experiencias nuevas en el aprendizaje de esta carrera, mis mejores compañeros y amigos de camino han sido mis padres, quienes me han alentado, se han sacrificado, han creído y confiado en todo momento en mí.

A ellos, a Dios, a mis hermanos Alexander y Elena y a toda mi familia, quiero dedicar esta investigación la cual ha sido no solo sacrificada, por el tiempo, las dificultades y la dedicación sino enriquecedora ya que abre las puertas de un nuevo conocimiento para mí y para aquellos que la lean.

Mi amor siempre para ustedes, gracias queridos padres Yolanda y Rolando.

AGRADECIMIENTO

Agradezco a Dios por darme la vida, voluntad y fuerzas para salir adelante en cada nuevo reto. Porque me dio el hermoso regalo de haber nacido en esta familia cristiana, quienes han fomentado el amor de la vida por los animales.

A mis padres, por la confianza que me brindaron siempre, por los sacrificios y paciencia, todo su esfuerzo no ha sido en vano, espero que la vida me conceda regresarles un poco de lo mucho que me han dado... ¡Los amo!

A mis hermanos, por creer en mí en todo momento, gracias por su cariño incondicional.

A toda mi familia, quienes me han demostrado todo su afecto y apoyo constante e invaluable, por compartir tantos momentos de alegría; que Dios los siga llenando de bendiciones y éxito, sepan que los llevo siempre en mi corazón.

De igual manera agradecer al Ing. Omar Barra Zapata, quien, con su experiencia, conocimientos y consejos, han logrado en mí, formarme como un gran profesional e investigador; quien no escatimó tiempo para orientarme y darme las directrices adecuadas para la culminación de esta preciada investigación.

Al Ing. Néstor Velásquez Linares por confiar en mí y brindarme todo su apoyo para la culminación de esta tesis y por brindarme su apoyo constante en todo momento.

Al Ing., Rolando Sosa Nizama, por todas sus enseñanzas a lo largo de mi camino, por recordarme a un labrador al mostrarme aquellos surcos en sus manos, por hablarme del peso de los años y decirme con firmeza “hijo” estas equivocado y por recordarme cuan eterno es el amor, al verlos felices juntos tantos años... junto a ella, la mujer que siempre estuvo a su lado.

Ismael Sosa Yarlequé

RESUMEN

El controlador lógico programable o simplemente PLC por sus siglas en inglés Programmable Logic Controller, es un equipo base de la automatización y control industrial aplicada a procesos y también a máquinas, incluso existen muchos controladores de gama baja que se utilizan para el control de piletas en plazas o en control de semáforos en grandes avenidas. Estos controladores son ampliamente utilizados en la industria, por ejemplo, en el control de procesos tipo batch como las embotelladoras, u otros procesos de manufactura.

Es así que, debido a su amplio uso, es imprescindible para un estudiante de ingeniería en el área de control, que pueda manejar alguna terminología para el manejo de este controlador.

Es por ello, que el presente proyecto tiene como finalidad el diseño de un controlador lógico programable que pueda ser fácilmente manipulable a nivel de hardware como de software, trabajando para ello con el lenguaje de programación por excelencia del PLC, el lenguaje escalera o Ladder, utilizando para ello un controlador ampliamente utilizado a nivel académico como el Arduino.

Palabras clave: PLC, Ladder, Arduino

ABSTRACT

The programmable logic controller or simply PLC by its acronym in English Programmable Logic Controller, is a basic equipment of automation and industrial control applied to processes and also to machines, there are even many low-end controllers that are used for the control of pools in squares or in control of traffic lights in large avenues. These controllers are widely used in the industry, for example in the control of batch processes such as bottling, or other manufacturing processes.

Thus, due to its wide use, it is essential for an engineering student in the control area, who can handle some terminology for the handling of this controller.

It is for this reason that the present project has the purpose of designing a programmable logic controller that can be easily manipulated at the hardware and software level, working for it with the programming language par excellence of the PLC, ladder or ladder language, using a controller widely used at the academic level as the Arduino

Keywords: PLC, Ladder, Arduino



UNIVERSIDAD NACIONAL DE PIURA

FACULTAD DE CIENCIAS



"AÑO DEL DIALOGO Y LA RECONCILIACIÓN NACIONAL"

ACTA DE SUSTENTACIÓN 055-2018-D-FC-UNP

Los Miembros del Jurado Calificador que suscriben, reunidos para evaluar la Tesis denominada **"DISEÑO DE UN PROGRAMADOR LÓGICO PROGRAMABLE USANDO MICROCONTROLADOR ATMEGA Y LENGUAJE LADDER PARA APLICACIONES DE LABORATORIO"** presentada por el señor Bachiller **SOSA YARLEQUÉ - ISMAEL**, con el asesoramiento del Ing. **Eduardo Omar Ávila Regalado**; oídas las observaciones y respuestas a las preguntas formuladas, y de conformidad al Reglamento de Tesis para obtener el Título Profesional en la Facultad de Ciencias, lo declaran:

APROBADO (X)

DESAPROBADO ()

Con la mención de:

Muy Bueno

(X) En consecuencia, queda en condición de ser ratificado por el Consejo de Facultad de Ciencias de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**.

(X) En consecuencia, queda en condición de ser ratificado por el Consejo Universitario de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**; después que el sustentante incorpore la sugerencia del Jurado Calificador.

Piura, 10 setiembre del 2018.

Dr. CARLOS ENRIQUE ARELLANO RAMÍREZ
PRESIDENTE DE JURADO DE TESIS

Ing. EDWIN ARNALDO OCAS INFANTE
SECRETARIO DE JURADO DE TESIS

MSc. FRANKLIN BARRA ZAPATA
VOCAL DE JURADO DE TESIS



INDICE

CAPÍTULO 1: PLANTEAMIENTO METODOLÓGICO			8
1.1.	DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA		8
1.2.	DEFINICIÓN Y LIMITACIONES DEL PROBLEMA		8
1.3.	FORMULACIÓN DEL PROBLEMA		9
1.4.	OBJETIVOS DE LA INVESTIGACIÓN		9
	1.4.1.	Objetivo general	9
	1.4.2.	Objetivos específicos	9
1.5.	JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN		10
1.6.	LIMITACIONES Y VIABILIDAD DE LA INVESTIGACIÓN		10
	1.6.1.	Limitaciones de la investigación	10
	1.6.2.	Viabilidad de la investigación	11
CAPÍTULO 2: MARCO TEÓRICO			12
2.1.	FUNDAMENTOS TEÓRICOS		12
	2.1.1.	ARDUINO	12
		2.1.1.1. Historia	15
		2.1.1.2. Hardware	17
		2.1.1.3. Especificaciones de Placas Arduino	24
		2.1.1.4. Bibliotecas de Arduino	26
	2.1.2.	CONTROLADOR LÓGICO PROGRAMABLE	27
		2.1.2.1. Campos de aplicación	31
		2.1.2.2. Modo de funcionamiento	32
		2.1.2.3. Estructura Externa	36
		2.1.2.4. Memorias	41
		2.1.2.5. Interfaces	45
		2.1.2.6. Módulo de Funciones Especiales	54
	2.1.3.	LENGUAJE LADDER	57
2.2.	ELABORACIÓN DE HIPÓTESIS		58
CAPÍTULO 3: METODOLOGÍA			59
3.1.	TIPO DE INVESTIGACIÓN, MÉTODO DE INVESTIGACIÓN		59

	3.1.1.	Tipo de investigación	59
	3.1.2.	Método de investigación	59
CAPÍTULO 4: DISEÑO DEL SISTEMA			60
4.1.	DESARROLLO DE HARDWARE		60
4.2.	SELECCIÓN DE LOS CIRCUITOS PARA LA FUENTE DE ALIMENTACIÓN		61
	4.2.1.	Reguladores de voltaje	61
4.3.	SELECCIÓN DE LA UNIDAD DE CONTROL DE PROCESOS		62
	4.3.1.	Lugar de la medición	60
	4.3.2.	Configuración de instalación	61
4.4.	DISEÑO DE ENTRADA ANALÓGICA		66
	4.4.1.	Etapa de acondicionamiento de señal	67
4.5.	DISEÑO DE LA SALIDA ANALÓGICA		68
	4.5.1.	Etapa de potencia	69
	4.5.2.	Circuito Integrado TL082	70
	4.5.3.	Circuito Integrado TIP120	67
4.6.	DISEÑO DE ENTRADA DIGITAL		72
	4.6.1.	Etapa de acondicionamiento de señal	72
	4.6.2.	Circuito integrado 4n25	73
4.7.	DISEÑO DE SALIDA DIGITAL		74
	4.7.1.	Relevador	75
4.8.	SOFTWARE DE PROGRAMACIÓN EN LENGUAJE ESCALERA		76
4.9.	DISEÑO DEL HARDWARE DEL PLC		79
4.10	PRUEBA DE SOFTWARE DE PROGRAMACIÓN EN LENGUAJE ESCALERA		86
CAPÍTULO 5: RESULTADOS Y CONCLUSIONES			88
CAPÍTULO 6: BIBLIOGRAFÍA			89
ANEXOS			91

CAPÍTULO 1: PLANTEAMIENTO METODOLÓGICO

1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA

El controlador lógico programable o simplemente PLC por sus siglas en inglés Programmable Logic Controller, es un equipo base de la automatización y control industrial aplicada a procesos y también a máquinas, incluso existen muchos controladores de gama baja que se utilizan para el control de piletas en plazas o en control de semáforos en grandes avenidas. Estos controladores son ampliamente utilizados en la industria, por ejemplo, en el control de procesos tipo batch como las embotelladoras, u otros procesos de manufactura.

Es así que, debido a su amplio uso, es imprescindible para un estudiante de ingeniería en el área de control, que pueda manejar alguna terminología para el manejo de este controlador.

Es por ello, que el presente proyecto tiene como finalidad el diseño de un controlador lógico programable que pueda ser fácilmente manipulable a nivel de hardware como de software, trabajando para ello con el lenguaje de programación por excelencia del PLC, el lenguaje escalera o Ladder.

Para cumplir con estos objetivos, se utilizarán microcontroladores ampliamente utilizables, como el Arduino, así como las herramientas de programación disponibles, y desde luego se tendrán que analizar los sistemas de acondicionamiento de señal para las entradas del PLC, así como las etapas de potencia para las salidas del PLC para manipular actuadores.

1.2. DEFINICIÓN Y LIMITACIONES DEL PROBLEMA

El uso de los controladores a nivel industrial, es muy importante, y lo que se pretende con la presente investigación es proporcionar al estudiante las herramientas necesarias para comprender la terminología que se usa al emplear estos dispositivos, así como el aprender a desenvolverse en programación en

escalera, que es la más usada a nivel industrial, esto sumado a que un controlador industrial básico puede llegar a tener un costo muy por encima de lo que podría pagar un estudiante en plena carrera universitaria, hace que el aprendizaje solo sea a nivel virtual sin “tocar” o manipular un equipo, por ello se puede decir que el problema principal radica en la falta de hardware y software necesarios para el aprendizaje a nivel de laboratorio del uso de un controlador lógico programable.

1.3. FORMULACIÓN DEL PROBLEMA

De acuerdo a lo antes visto podemos formular el problema en forma de pregunta cómo se muestra a continuación:

¿Será posible diseñar un programador lógico programable usando microcontrolador ATMEGA y lenguaje LADDER para aplicaciones de laboratorio?

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. Objetivo general

- Diseñar de un programador lógico programable usando microcontrolador ATMEGA y lenguaje LADDER para aplicaciones de laboratorio.

1.4.2. Objetivos específicos

- Diseñar la placa de desarrollo basada en el microcontrolador ATMEGA para la aplicación del controlador lógico programable.
- Diseñar los circuitos de acondicionamiento de señal para las señales de entrada analógica y digital.
- Diseñar los circuitos de etapa de potencia para las señales de salida analógica y digital.

- Seleccionar los dispositivos necesarios para el funcionamiento del controlador lógico programable.
- Realizar pruebas a nivel de hardware y software del controlador lógico programable.

1.5. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

De acuerdo a las enseñanzas recibidas en nuestra carrera universitaria, además de tener experiencias con el diseño de circuitos, podemos decir que podremos desarrollar el diseño del presente proyecto. Otro punto que es muy importante es que existe suficiente información para desarrollar diversas aplicaciones con los microcontroladores Arduino.

La importancia de la investigación radica en que el diseño del sistema en general permitirá que el estudiante de ingeniería electrónica tenga las herramientas necesarias para el aprendizaje en el manejo de un controlador lógico programable a nivel de hardware y software.

1.6. LIMITACIONES Y VIABILIDAD DE LA INVESTIGACIÓN

1.6.1. Limitaciones de la investigación

Para el desarrollo del presente proyecto se presume que presenten las siguientes limitantes:

- Lo ideal sería implementar un hardware lo más parecido a un PLC industrial, pero la limitante es el costo del desarrollo de la matriz para la elaboración de la caja industrial.
- Por temas de costo, nos abocaremos a utilizar software de preferencia con licencias gratuitas, ya que el objetivo es tener un equipo de bajo coste.

1.6.2. Viabilidad de la investigación

Como nuestra investigación será un diseño, se revisará información de diseños de circuitos de PLC existentes, tomando en cuenta que sea información abierta o sin restricciones de propiedad intelectual.

CAPÍTULO 2: MARCO TEÓRICO

3.1. FUNDAMENTOS TEÓRICOS

3.1.1. ARDUINO

De acuerdo a lo que se indica en la página oficial de Arduino (2017), Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Adicionalmente Arduino puede capturar señales físicas del entorno mediante la recepción de entradas provenientes de una variedad de sensores (temperatura, presión, intensidad luminosa, nivel, etc.) y puede controlar su alrededor mediante el control de actuadores (como luces, motores, relés, entre otros); para ello el Arduino posee entradas y salidas analógicas y digitales, para que se pueda conectar con sensores u actuadores.

El microcontrolador de la placa, se programa usando el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Las placas se pueden ensamblar a mano o encargarlas pre-ensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica Prix en 2006.

Existe gran variedad de microcontroladores y placas de aprendizaje con microcontroladores, como por ejemplo el microcontrolador PIC de Microchip, que tienen diversas placas de entrenamiento y de aprendizaje que incluso utilizan lenguajes de programación Basic (como se muestra en el libro de Barra y otros 2011). El detalle de muchos estos lenguajes de programación de alto nivel, es que son compiladores que implican un costo importante tomando en

cuenta que se aplica para aprendizaje. Por ello y como se indica en la página oficial de Arduino (2017), este dispositivo ofrece algunas ventajas para los interesados en desarrollar proyectos con microcontroladores, los cuales se detalla a continuación:

- **Barato:** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de \$50.
- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux, en cambio, la mayoría de los sistemas microcontroladores están limitados a Windows.
- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.



Figura 2.1. Arduino Uno revisión 3, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-uno-rev3>

Como se mencionó en párrafos anteriores, para el uso de un Arduino, se necesita de un hardware y un software, el hardware es en una placa de circuito impreso con un microcontrolador, que como ya se mencionó, es usualmente un Atmel AVR, con sus puertos digitales y analógicos de entrada o salida, los cuales pueden conectarse a placas de expansión, por ejemplo de sensores o actuadores, que amplían los funcionamientos de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

Por otro lado, se necesita del software que es un entorno de desarrollo (IDE) basado en el entorno de processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa, tome en cuenta que este software de arranque viene previamente almacenado en la placa. El microcontrolador de la placa se programa mediante un computador, usando una comunicación serie mediante un convertidor de niveles RS-232 a TTL, a través del puerto USB.

Como se va a mencionar posteriormente, la primera placa Arduino fue introducida en 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales. Buscaba desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se

incorporaron nuevos modelos de placas de desarrollo que usan microcontroladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura, por lo cual tampoco lo es su set de instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.

Las placas Arduino están disponibles de dos formas: ensambladas o en forma de kits "Hazlo tú mismo" (por sus siglas en inglés "DIY"). Los esquemas de diseño del hardware están disponibles bajo licencia libre, con lo que se permite que cualquier persona pueda crear su propia placa Arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que, alrededor de 300 000 placas Arduino habían sido producidas comercialmente y en el año 2013 estimó que alrededor de 700 000 placas oficiales de la empresa Arduino estaban en manos de los usuarios (Wikipedia 2017).

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data, etc. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW. El entorno de desarrollo integrado es software libre, y por tanto se puede descargar y modificar gratuitamente; utiliza la licencia Creative Commons, con copyleft y derechos de atribución al autor.

Arduino como herramienta educativa es muy útil y efectiva. Existen diferentes web con recursos, tutoriales, trucos y ejercicios, además de existir tutoriales oficiales de Arduino. La plataforma en sí misma tiene una gran comunidad a su alrededor donde se puede encontrar muchos recursos, desde tutoriales para iniciarse desde cero hasta aquellos destinados a usuarios más avanzados.

3.1.1.1.Historia

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). En ese tiempo, los estudiantes usaban el

microcontrolador BASIC Stamp, cuyo costo era de 100 dólares estadounidenses, lo que se consideraba demasiado costoso para ellos. Por aquella época, uno de los fundadores de Arduino, Massimo Banzi, daba clases en Ivrea. (Wikipedia 2017).

El nombre del proyecto viene del nombre del Bar di Re Arduino (Bar del Rey Arduino) donde Massimo Banzi pasaba algunas horas. El rey Arduino fue rey de Italia entre los años 1002 y 1014. En la creación de este proyecto contribuyó el estudiante colombiano Hernando Barragán, quien desarrolló la tarjeta electrónica Wiring, el lenguaje de programación y la plataforma de desarrollo. Una vez concluida dicha plataforma, los investigadores trabajaron para hacerlo más ligero, más económico y disponible para la comunidad de código abierto (hardware y código abierto). El instituto finalmente cerró sus puertas, así que los investigadores, entre ellos el español David Cuartielles, promovieron la idea. Banzi afirmaría años más tarde, que el proyecto nunca surgió como una idea de negocio, sino como una necesidad de subsistir ante el inminente cierre del Instituto de diseño Interactivo IVREA. Es decir, que al crear un producto de hardware abierto, éste no podría ser embargado.

Posteriormente, Google colaboró en el desarrollo del Kit Android ADK (Accessory Development Kit), una placa Arduino capaz de comunicarse directamente con teléfonos móviles inteligentes bajo el sistema operativo Android para que el teléfono controle luces, motores y sensores conectados de Arduino. (Accessory Development Kit 2012 Guide, 2012)

Para la producción en serie de la primera versión se tomó en cuenta que el coste no fuera mayor de 30 euros, que fuera ensamblado en una placa de color azul, debía ser Plug and Play y que trabajara con todas las plataformas informáticas tales como MacOSX, Windows y GNU/Linux. Las primeras 300 unidades se las dieron a los alumnos del Instituto IVREA, con el fin de que las probaran y empezaran a diseñar sus primeros prototipos.

En el año 2005, se incorporó al equipo el profesor Tom Igoe, que había trabajado en computación física, después de que se enterara del mismo a través de internet. Igoe ofreció su apoyo para desarrollar el proyecto a gran escala y

hacer los contactos para distribuir las tarjetas en territorio estadounidense. En la feria Maker Fair de 2011 se presentó la primera placa Arduino de 32 bits para realizar tareas más pesadas.

3.1.1.2.Hardware

De acuerdo a lo que se indica en Wikipedia (2017), los modelos de Arduino se categorizan en placas de desarrollo, placas de expansión (shields), kits, accesorios e impresoras 3d.

- Placas
 - Arduino Galileo
 - Arduino Uno
 - Arduino Leonardo
 - Arduino Due
 - Arduino Yún
 - Arduino Tre (En Desarrollo)
 - Arduino Zero
 - Arduino Micro
 - Arduino Esplora
 - Arduino Mega ADK
 - Arduino Ethernet
 - Arduino Mega 2560
 - Arduino Robot
 - Arduino Mini

- Arduino Nano
- LilyPad Arduino Simple
- LilyPad Arduino SimpleSnap
- LilyPad Arduino
- LilyPad Arduino USB
- Arduino Pro Mini
- Arduino Fio
- Arduino Pro
- Arduino MKR1000/Genuino MKR1000
- Arduino MICRO/Genuino MICRO
- Arduino 101/Genuino 101
- Arduino Gemma

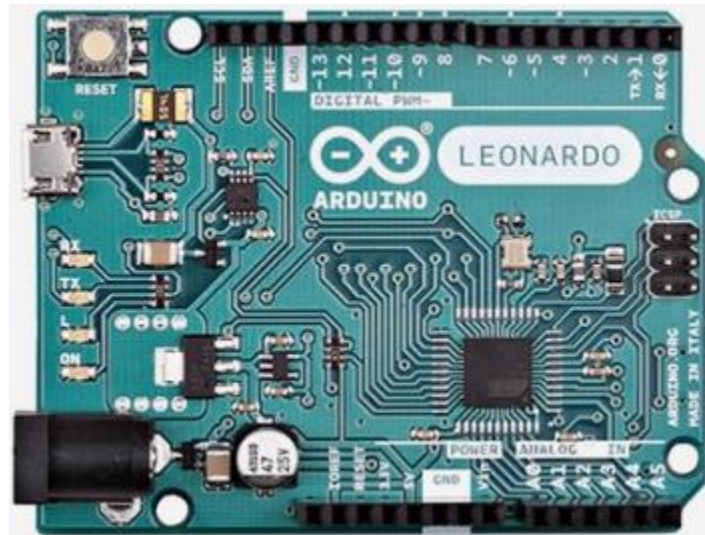


Figura 2.2. Placa Arduino Leonardo, cortesía de la página oficial de Arduino en:
<https://store.arduino.cc/usa/arduino-leonardo-with-headers>

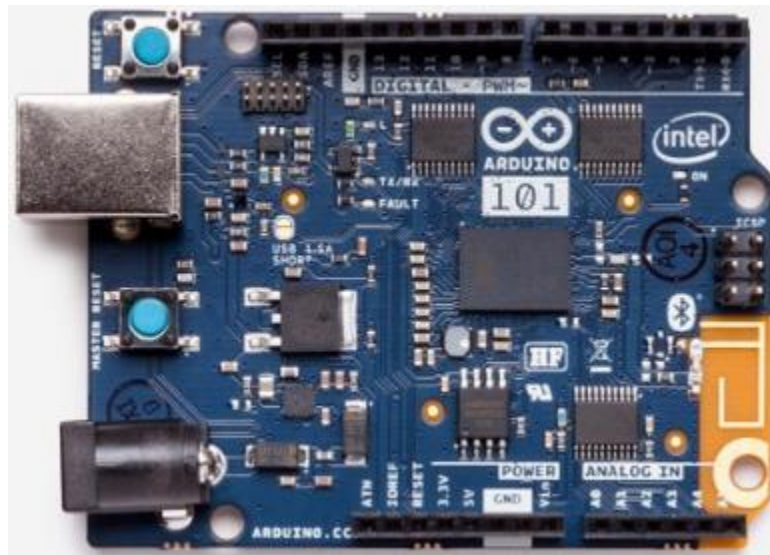


Figura 2.3. Placa Arduino 101, cortesía de la página oficial de Arduino en:
<https://store.arduino.cc/usa/arduino-101>



Figura 2.4. Placa Arduino Esplora, cortesía de la página oficial de Arduino en:
<https://store.arduino.cc/usa/arduino-esplora>



Figura 2.5. Placa Arduino Micro, cortesía de la página oficial de Arduino en:
<https://store.arduino.cc/usa/arduino-micro>

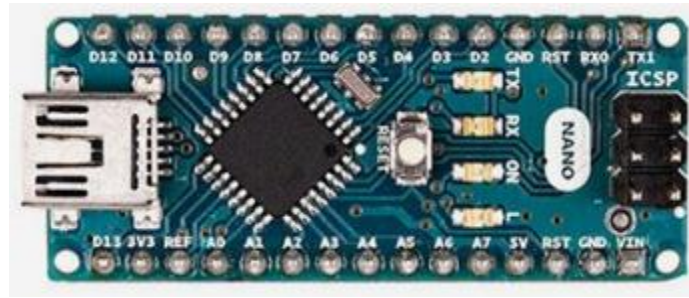


Figura 2.6. Placa Arduino Nano, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-nano>



Figura 2.7. Placa Arduino Mega, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

- Placas de expansión (shields)
 - Arduino GSM Shield
 - Arduino Ethernet Shield
 - Arduino WiFi Shield
 - Arduino Wireless SD Shield
 - Arduino USB Host Shield
 - Arduino Motor Shield
 - Arduino Wireless Proto Shield
 - Arduino Proto Shield



Figura 2.8. Placa de expansión Arduino Motor Shield REV3, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-motor-shield-rev3>



Figura 2.9. Placa de Expansión Arduino Gsm Shield 2 (Integrated Antenna), cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-gsm-shield-2-integrated-antenna>

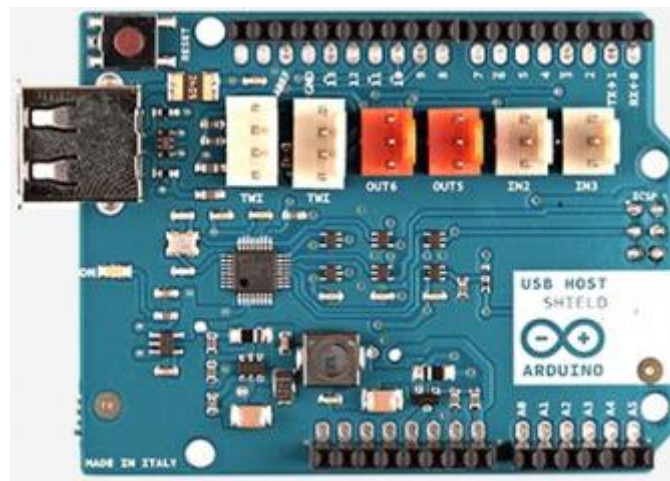


Figura 2.10. Placa de Expansión Arduino Usb Host Shield, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-usb-host-shield>

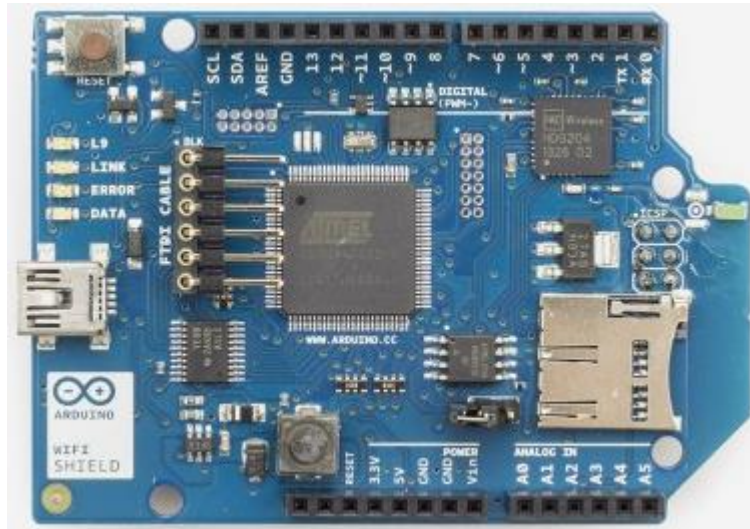


Figura 2.11. Placa de Expansión Arduino Wifi Shield, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-wifi-shield>

- Kits
 - The Arduino Starter Kit, Arduino Materia 101.



Figura 2.12. Kit Arduino Starter Kit Multi-Language, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-starter-kit>

- Accesorios

- Pantalla LCD TFT, Adaptador USB/Serie y MiniUSB/Serie, Arduino ISP.



Figura 2.13. Pantalla de cristal líquido para Arduino, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-lcd-screen>

- Impresoras 3d
 - Arduino Materia 101.



Figura 2.14. Impresora 3D Arduino Materia 101 - Assembled, cortesía de la página oficial de Arduino en: <https://store.arduino.cc/usa/arduino-starter-kit>

3.1.1.3. Especificaciones de Placas Arduino

A continuación mostramos la tabla 2.1, donde se muestran las características esenciales de las placas Arduino más conocidas.

Tabla 2.1. Especificaciones generales de las placas Arduino. Tabla obtenida de Wikipedia 2017.

Modelo	Microcontrolador	Voltaje de entrada	Voltaje del sistema	Frecuencia de reloj	Entradas/salidas digitales	Entradas analógicas	PWM	UART	Memoria flash	Cargador	Interfaz de programación
Arduino Due	AT91SAM3X8E	5-12V	3,3V	84MHz	54*	12	12	4	512Kb	Due	Nativa USB
Arduino Leonardo	ATmega32U4	7-12V	5V	16MHz	20*	12	7	1	32Kb	Leonardo	Nativa USB
Arduino Uno - R3	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB vía ATmega16U2
RedBoard	ATmega328	7-15V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB vía FTDI
Arduino Uno SMD (descontinuado)	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB vía ATmega8U2
Arduino Uno (descontinuado)	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB vía ATmega8U2
Arduino Duemilanova (descontinuado)	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	AtmegaBOOT	USB vía FTDI
Arduino Bluetooth (descontinuado)	ATmega328	1,2-5,5V	5V	16MHz	14	6	6	1	32Kb	AtmegaBOOT	Serial Bluetooth
Arduino Pro 3,3V/8MHz	ATmega328	3,35-12V	3,3V	8MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
Arduino Pro 5V/16MHz	ATmega328	5-12V	5V	16MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
Ethernet Pro (descontinuado)	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
Arduino Mega 2560 R3	ATmega2560	7-12V	5V	16MHz	54	16	14	4	256Kb	STK500 v2	USB vía ATmega16U2
Arduino Mega 2560 (descontinuado)	ATmega2560	7-12V	5V	16MHz	54	16	14	4	256Kb	STK500 v2	USB vía ATmega8U2
Arduino Mega	ATmega1280	7-12V	5V	16MHz	54	16	14	4	128Kb	STK500 v2	USB vía FTDI

(descontin uado)											
Mega Pro 3.3V	ATmega2560	3,3-12V	3,3V	8MHz	54	16	14	4	256Kb	STK500v2	Cabecera compatible con FTDI
Mega Pro 5V	ATmega2560	5-12V	5V	16MHz	54	16	14	4	256Kb	STK500v2	Cabecera compatible con FTDI
Arduino Mini 04 (descontin uado)	ATmega328	7-9V	5V	16MHz	14	6	8	1	32Kb	AtmegaBOOT	Cabecera Serial
Arduino Mini 05	ATmega328	7-9V	5V	16MHz	14	6	8	1	32Kb	AtmegaBOOT	Cabecera Serial
Arduino Pro Mini 3.3V/8MHz	ATmega328	3,35-12V	3,3V	8MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
Arduino Pro Mini 5V/16MHz	ATmega328	5-12V	5V	16MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
Arduino Fio	ATmega328P	3,35-12V	3,3V	8MHz	14	8	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI o inalámbrica vía XBee ¹
Mega Pro Mini 3.3V	ATmega2560	3,3-12V	3,3V	8MHz	54	16	14	4	256Kb	STK500v2	Cabecera compatible con FTDI
Pro Micro 5V/16MHz	ATmega32U4	5-12V	5V	16MHz	12	4	5	1	32Kb	DiskLoader	Nativa USB
Pro Micro 3.3V/8MHz	ATmega32U4	3,35-12V	3,3V	8MHz	12	4	5	1	32Kb	DiskLoader	Nativa USB
LilyPad Arduino 328 Main Board	ATmega328	2,7-5,5V	3,3V	8MHz	14	6	6	1	32Kb	AtmegaBOOT	Cabecera compatible con FTDI
LilyPad Arduino Simple Board	ATmega328	2,7-5,5V	3,3V	8MHz	9	4	5	0 ²	32Kb	AtmegaBOOT	Cabecera compatible con FTDI

Adicionalmente mostramos las características básicas de los controladores ATmega, que se utilizan mayormente en las placas Arduino, los cuales son: ATmega168, ATmega328 y ATmega1280. Estas características se muestran en la tabla 2.2.

Tabla 2.1. Especificaciones generales de los microcontroladores ATmega: ATmega168, ATmega328 y ATmega1280. Tabla obtenida de Wikipedia 2017.

	ATmega168	ATmega328	ATmega1280
Voltaje operativo	5 V	5 V	5 V
Voltaje de entrada recomendado	7-12 V	7-12 V	7-12 V
Voltaje de entrada límite	6-20 V	6-20 V	6-20 V
Entradas y salidas digitales	14 (6 proporcionan PWM)	14 (6 proporcionan PWM)	54 (14 proporcionan PWM)
Entradas analógicas	6	6	16
Intensidad de corriente	40 mA	40 mA	40 mA
Memoria Flash	16KB (2KB reservados para el bootloader)	32KB (2KB reservados para el bootloader)	128KB (4KB reservados para el bootloader)
SRAM	1 KB	2 KB	8 KB
EEPROM	512 bytes	1 KB	4 KB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz

3.1.1.4. Bibliotecas de Arduino

Las bibliotecas estándar que ofrece Arduino son las siguientes:

- **Serial:** Lectura y escritura por el puerto serie.
- **EEPROM:** Lectura y escritura en el almacenamiento permanente. Las instrucciones utilizadas son: read(), write()
- **Ethernet:** Conexión a Internet mediante “Arduino Ethernet Shield“. Puede funcionar como servidor que acepta peticiones remotas o como cliente. Se permiten hasta cuatro conexiones simultáneas. Las instrucciones usadas son los siguientes: Servidor: Server(), begin(), available(), write(), print(), println(). Cliente: Client(), connected(), connect(), write(), print(), println(), available(), read(), flush(), stop().
- **Firmata:** Es una biblioteca de comunicación con aplicaciones informáticas utilizando el protocolo estándar del puerto serie.
- **LiquidCrystal:** Control de LCDs con chipset Hitachi HD44780 o compatibles. La biblioteca soporta los modos de 4 y 8 bits.
- **Servo:** Biblioteca para el control de servo motores. A partir de la versión 0017 de Arduino la biblioteca soporta hasta 12 motores en la mayoría de las

placas Arduino y 48 en la Arduino Mega. Estas son las instrucciones utilizadas: `attach()`, `write()`, `writeMicroseconds()`, `read()`, `attached()`, `detach()`

- **SoftwareSerial:** Comunicación serie en contactos digitales.³⁰ Por defecto Arduino incluye comunicación sólo en los contactos 0 y 1 pero gracias a esta biblioteca puede realizarse esta comunicación con los restantes.
- **Stepper:** Control de motores paso a paso unipolares o bipolares. Las instrucciones usadas son los siguientes: `Stepper(steps, pin1, pin2)`, `Stepper(steps, pin1, pin2, pin3, pin4)`, `setSpeed(rpm)`, `step(steps)`
- **Wire:** Envío y recepción de datos sobre una red de dispositivos o sensores mediante Two Wire Interface (TWI/I2C). Las bibliotecas Matrix y Sprite de Wiring son totalmente compatibles con Arduino y sirven para manejo de matrices de ledes. También se ofrece información sobre diversas bibliotecas desarrolladas por diversos colaboradores que permiten realizar muchas tareas.
- **Creación de bibliotecas:** Los usuarios de Arduino tienen la posibilidad de escribir sus propias bibliotecas. Ello permite disponer de código que puede reutilizarse en otros proyectos, mantener el código fuente principal separado de las bibliotecas y la organización de los programas construidos es más clara.

3.1.2. CONTROLADOR LÓGICO PROGRAMABLE

De acuerdo a lo indicado en Wikipedia (2017), un controlador lógico programable, más conocido por sus siglas en inglés PLC (Programmable Logic Controller) o por autómatas programables, es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.

Hasta no hace mucho tiempo el control de procesos industriales se venía haciendo de forma cableada por medio de contactores y relees. Al operario que se encontraba a cargo de este tipo de instalaciones, se le exigía tener altos conocimientos técnicos para poder realizarlas y posteriormente mantenerlas. Además cualquier variación en el proceso suponía modificar físicamente gran

parte de las conexiones de los montajes, siendo necesario para ello un gran esfuerzo técnico y un mayor desembolso económico.

En la actualidad no se puede entender un proceso complejo de alto nivel desarrollado por técnicas cableadas. El ordenador y los Controladores Lógicos Programables han intervenido de forma considerable para que este tipo de instalaciones se hayan visto sustituidas por otras controladas de forma programada.

El Controlador Lógico Programable (PLC) nació como solución al control de circuitos complejos de automatización. Por lo tanto se puede decir que un PLC no es más que un aparato electrónico que sustituye los circuitos auxiliares o de mando de los sistemas automáticos. A él se conectan los captadores (finales de carrera, pulsadores, etc.) por una parte, y los actuadores (bobinas de contactores, lámparas, pequeños receptores, etc.) por otra.

Los PLC se introdujeron por primera vez en la industria en 1960 aproximadamente. La razón principal de tal hecho fue la necesidad de eliminar el gran costo que se producía al reemplazar el complejo sistema de control basado en relees y contactores. Bedford Associates propuso algo denominado Controlador Digital Modular (MODICON, Modular Digital Controller) a un gran fabricante de coches. Otras compañías propusieron a la vez esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8. El MODICON 084 (Schneider) resultó ser el primer PLC del mundo en ser producido comercialmente.

El problema de los relés era que cuando los requerimientos de producción cambiaban también lo hacía el sistema de control. Esto comenzó a resultar bastante caro cuando los cambios fueron frecuentes. Dado que los relés son dispositivos mecánicos y poseen una vida limitada se requería un estricto mantenimiento planificado. Por otra parte, a veces se debían realizar conexiones entre cientos o miles de relés, lo que implicaba un enorme esfuerzo de diseño y mantenimiento.

Los "nuevos controladores" debían ser fácilmente programables por ingenieros de planta o personal de mantenimiento. El tiempo de vida debía ser largo y los cambios en el programa tenían que realizarse de forma sencilla. Finalmente se

imponía que trabajaran sin problemas en entornos industriales adversos. La solución fue el empleo de una técnica de programación familiar y reemplazar los relés mecánicos por relés de estado sólido.

A mediados de los 70 las tecnologías dominantes de los PLC eran máquinas de estado secuencial y CPU basadas en desplazamiento de bit. Los microprocesadores convencionales cedieron la potencia necesaria para resolver de forma rápida y completa la lógica de los pequeños PLC. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo.

Las habilidades de comunicación comenzaron a aparecer en 1973 aproximadamente. El primer sistema fue el bus Modicon (Modbus). El PLC podía ahora dialogar con otros PLC y en conjunto podían estar aislados de las máquinas que controlaban. También podían enviar y recibir señales de tensión variables, entrando en el mundo analógico. Desafortunadamente, la falta de un estándar acompañado con un continuo cambio tecnológico ha hecho que la comunicación de PLC sea un maremagnum de sistemas físicos y protocolos incompatibles entre si. No obstante fue una gran década para los PLC.

En los 80 se produjo un intento de estandarización de las comunicaciones con el protocolo MAP (Manufacturing Automation Protocol) de General Motor's. También fue un tiempo en el que se redujeron las dimensiones del PLC y se pasó a programar con programación simbólica a través de ordenadores personales en vez de los clásicos terminales de programación. Hoy día el PLC más pequeño es del tamaño de un simple rele.

Los 90 han mostrado una gradual reducción en el número de nuevos protocolos, y en la modernización de las capas físicas de los protocolos más populares que sobrevivieron a los 80. El último estándar (IEC 1131-3) intenta unificar el sistema de programación de todos los PLC en un único estándar internacional. Ahora disponemos de PLC's que pueden ser programados en diagramas de bloques, lista de instrucciones y texto estructurado al mismo tiempo.

Los PC están comenzando a reemplazar al PLC en algunas aplicaciones, incluso la compañía que introdujo el Modicon 084 ha cambiado al control basado en PC. Por lo cual, no sería de extrañar que en un futuro no muy lejano el PLC

desaparezca frente al cada vez más potente PC, debido a las posibilidades que los ordenadores pueden proporcionar

Actualmente, los PLC son utilizados en muchas industrias y máquinas. A diferencia de las computadoras de propósito general, el PLC está diseñado para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. Los programas para el control de funcionamiento de la máquina se suelen almacenar en baterías, copia de seguridad o en memorias no volátiles. Un PLC es un ejemplo de un sistema de tiempo real «duro», donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no producirá el resultado deseado.



Figura 2.15 Fotografía de un PLC Siemens Simatic S7-400, de izquierda a derecha: fuente de alimentación (PS), CPU, módulo de interfaz (IM) y el procesador de comunicaciones (CP).

Cortesía de Wikipedia 2017.

Entre las principales ventajas tenemos:

- Menor tiempo de elaboración de proyectos, debido a que no es necesario dibujar el esquema de contactos.

- Posibilidad de añadir modificaciones sin costo añadido en otros componentes.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra de la instalación.
- Mantenimiento económico.
- Posibilidad de gobernar varias máquinas con el mismo PLC
- Menor tiempo de puesta en funcionamiento.

Si el PLC queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

Entre los inconvenientes podemos citar el Adiestramiento de técnicos y su costo. Al día de hoy estos inconvenientes se van haciendo cada vez menores, ya que todos los PLC comienzan a ser más sencillos de programar, algunos se los programa con símbolos. En cuanto al costo tampoco hay problema, ya que hay Controladores Lógicos Programables para todas las necesidades y a precios ajustados.

3.1.2.1.Campos de aplicación

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc., por tanto, su aplicación abarca desde procesos de fabricación industriales de cualquier tipo a transformaciones industriales, control de instalaciones, etc.

Sus reducidas dimensiones, la extremada facilidad de su montaje, la posibilidad de almacenar los programas para su posterior y rápida utilización, la modificación o alteración de los mismos, etc., hace que su eficacia se aprecie fundamentalmente en procesos en que se producen necesidades tales como:

- Espacio reducido.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.
- Chequeo de programación centralizada de las partes del proceso.
- Maniobra de máquinas.
- Maniobra de instalaciones.
- Señalización y control.
- Chequeo de Programas.
- Señalización del estado de procesos.

Tal y como dijimos anteriormente, esto se refiere a los Controlador Lógico Programable industriales, dejando de lado los pequeños PLC para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, como la puerta de un cochera o las luces de la casa).

3.1.2.2. Modo de funcionamiento

Los Controladores Lógicos Programables son máquinas secuenciales que ejecutan correlativamente las instrucciones indicadas en el programa de usuario almacenado en su memoria, generando unas órdenes o señales de mando a partir de las señales de entrada leídas de la planta (aplicación): al detectarse cambios en las señales, el autómata reacciona según el programa hasta obtener las órdenes de salida necesarias. Esta secuencia se ejecuta continuamente para conseguir el control actualizado del proceso.

La secuencia básica de operación del autómata se puede dividir en tres fases principales:

- Lectura de señales desde la interfaz de entradas.
- Procesado del programa para obtención de las señales de control.

- Escritura de señales en la interfaz de salidas.

A fin de optimizar el tiempo, la lectura y escritura de las señales se realiza a la vez para todas las entradas y salidas; Entonces, las entradas leídas de los módulos de entrada se guardan en una memoria temporal (Imagen entradas). A esta acude la CPU en la ejecución del programa, y según se va obteniendo las salidas, se guardan en otra memoria temporal (imagen de salida). Una vez ejecutado el programa completo, estas imágenes de salida se transfieren todas a la vez al módulo de salida.

Ciclo de funcionamiento

El funcionamiento del Controlador Lógico Programable es, salvo el proceso inicial que sigue a un Reset, de tipo secuencial y cíclico, es decir, las operaciones tienen lugar una tras otra, y se van repitiendo continuamente mientras el autómatas esté bajo tensión.

La figura 2.16 muestra esquemáticamente la secuencia de operaciones que ejecuta el autómatas, siendo las operaciones del ciclo de operación las que se repiten indefinidamente. El ciclo de funcionamiento se divide en dos partes como se puede observar en el esquema de diagrama de la figura 2.16 llamados Proceso Inicial y Ciclo de Operación.

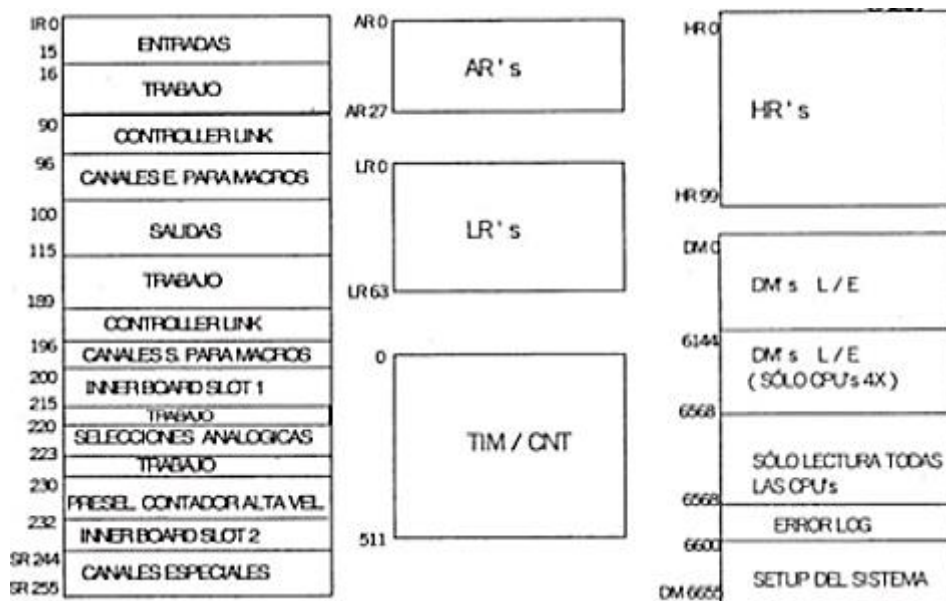


Figura 2.16 Ciclo de funcionamiento de un PLC.

Proceso inicial

Como se muestra en la figura, antes de entrar en el ciclo de operación el autómata realiza una serie de acciones comunes, que tratan fundamentalmente de inicializar los estados del mismo y chequear el hardware. Estas rutinas de chequeo, incluidas en el programa monitor ROM, comprueban:

- El bus de conexiones de las unidades de E/S.
- El nivel de la batería, si esta existe.
- La conexión de las memorias internas del sistema.
- El módulo de memoria exterior conectado, si existe.

Si se encontrara algún error en el chequeo, se activaría el LED de error y quedaría registrado el código del error.

Comprobadas las conexiones, se inicializan las variables internas:

- Se ponen a OFF las posiciones de memoria interna (excepto las mantenidas o protegidas contra perdidas de tensión)
- Se borran todas las posiciones de memoria imagen E/S.
- Se borran todos los contadores y temporizadores (excepto los mantenidos o protegidos contra perdidas de tensión).

Transcurrido el Proceso Inicial y si no han aparecido errores el autómata entra en el Ciclo de Operación.

Ciclo de operación

Este ciclo puede considerarse dividido en tres bloques:

- Proceso Común.
- Ejecución del programa.
- Servicio a periféricos.

Proceso común:

En este primer bloque se realizan los chequeos cíclicos de conexiones y de memoria de programa, protegiendo el sistema contra:

- Errores de hardware (conexiones E/S, ausencia de memoria de programa, etc.).
- Errores de sintaxis (programa imposible de ejecutar).

El chequeo cíclico de conexiones comprueba los siguientes puntos:

- Niveles de tensión de alimentación.
- Estado de la batería si existe.
- Buses de conexión con las interfaces.

El chequeo de la memoria de programa comprueba la integridad de la misma y los posibles errores de sintaxis y gramática:

- Mantenimiento de los datos, comprobados en el "checksum".
- Existencia de la instrucción END de fin de programa.
- Estructura de saltos y anidamiento de bloque correctas.
- Códigos de instrucciones correctas.

Ejecución del programa:

En este segundo bloque se consultan los estados de las entradas y de las salidas y se elaboran las órdenes de mando o de salida a partir de ellos.

El tiempo de ejecución de este bloque de operaciones es la suma del:

- Tiempo de acceso a interfaces de E/S.
- Tiempo de escrutación de programa.

Y a su vez esto depende, respectivamente de:

- Número y ubicación de las interfaces de E/S.
- Longitud del programa y tipo de CPU que lo procesa.

Servicio a periféricos

Este tercer y último bloque es únicamente atendido si hay pendiente algún intercambio con el exterior. En caso de haberlo, la CPU le dedica un tiempo limitado, de 1 a 2 ms, en atender el intercambio de datos. Si este tiempo no fuera suficiente, el servicio queda interrumpido hasta el siguiente ciclo.

Tiempo de ejecución y control en tiempo real

El tiempo total que el Controlador Lógico Programable emplea para realizar un ciclo de operación se llama tiempo de ejecución de ciclo de operación o más sencillamente tiempo de ciclo "Scan time".

Dicho tiempo depende de:

- El número de E/S involucradas.
- La longitud del programa usuario.
- El número y tipo de periféricos conectados al autómata.

Los tiempos totales de ciclos son entonces la suma de tiempos empleados en realizar las distintas operaciones del ciclo

- Autodiagnóstico (Proceso común).
- Actualización de E/S (Ejecución del programa).
- Ejecución de programa. (Ejecución del programa).
- Servicio a periféricos. (Servicio a periféricos).

3.1.2.3. Estructura Externa

El término estructura externa o configuración externa de un Controlador Lógico programable industrial se refiere al aspecto físico exterior del mismo, bloques o elementos en que está dividido.

Actualmente son tres las estructuras más significativas que existen en el mercado:

- Estructura compacta.
- Estructura semimodular. (Estructura Americana).
- Estructura modular. (Estructura Europea).

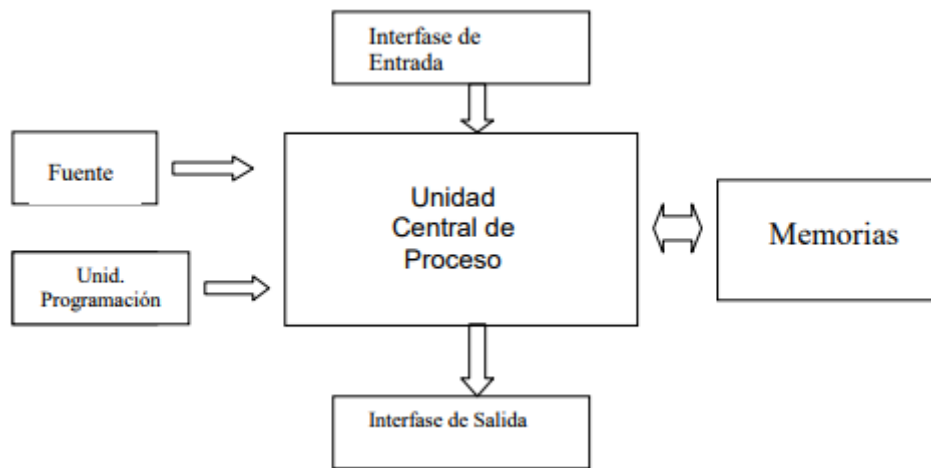


Figura 2.17 Estructura de un PLC indicando sus partes.

Estructura compacta

Este tipo de Controlador Lógico Programable se distingue por presentar en un solo bloque todos sus elementos, esto es, fuente de alimentación, CPU, memorias, entradas/salidas, etc.

Son los PLC de gama baja o nanoautómatas los que suelen tener una estructura compacta.

Su potencia de proceso suele ser muy limitada dedicándose a controlar máquinas muy pequeñas o cuadros de mando.

Estructura semimodular

Se caracteriza por separar las E/S del resto del Controlador Lógico Programable, de tal forma que en un bloque compacto están reunidas las CPU, memoria de usuario o de programa y fuente de alimentación y separadamente las unidades de E/S.

Son los Controlador Lógico Programable de gama media los que suelen tener una estructura semimodular (Americana).

Estructura modular

Su característica principal es la de que existe un módulo para cada uno de los diferentes elementos que componen el PLC como puede ser una fuente de alimentación, CPU, E/S, etc.

La sujeción de los mismos se hace por riel DIN, placa perforada o sobre RACK, en donde van alojado el BUS externo de unión de los distintos módulos que componen.

Son los PLC de gama alta los que suelen tener una estructura modular, que permiten una gran flexibilidad en su constitución.

Unidad de Programación

Es el conjunto de medios, hardware y software mediante los cuales el programador introduce y depura sobre las secuencias de instrucciones (en uno u otro lenguaje) que constituyen el programa a ejecutar.

Esta puede estar constituida por un teclado pequeño adosable al controlador, donde cada tecla responderá a un elemento del circuito/programa a desarrollar.

Fuente de Alimentación

La fuente de alimentación proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos del sistema.

La alimentación a la CPU puede ser de continua a 24 Vcc, tensión muy frecuente en cuadros de distribución, o en alterna a 110 / 220 Vca. En cualquier caso es la propia CPU la que alimenta las interfaces conectadas a través del bus interno.

La alimentación a los circuitos E/S puede realizarse, según tipos, en alterna a 48/110/220 Vca o en continua a 12/24/48 Vcc.

La fuente de alimentación del Controlador Lógico Programable puede incorporar una batería de reserva, que se utiliza para el mantenimiento de algunas posiciones internas y del programa usuario en memoria RAM, o cuando falla la alimentación o se apaga el Controlador Lógico Programable.

Unidad Central de Proceso

La CPU (Central Processing Unit) es la parte inteligente del sistema. Interpreta las instrucciones del programa de usuario y consulta el estado de las entradas.

Dependiendo de dichos estados y del programa, ordena la activación de las salidas deseadas.

La CPU está constituida por los siguientes elementos:

- Procesador
- Memoria monitor del sistema
- Circuitos auxiliares

Procesador

Está constituido por el microprocesador, el reloj (generador de onda cuadrada) y algún chip auxiliar.

El microprocesador es un circuito integrado (chip), que realiza una gran cantidad de operaciones, que podemos agrupar en:

- Operaciones de tipo lógico.
- Operaciones de tipo aritmético.
- Operación de lectura y modificación de datos.
- Operaciones de entrada - salida
- Operaciones de control de la transferencia de la información dentro del autómata.

Para que el microprocesador pueda realizar todas estas operaciones está dotado de unos circuitos internos que son los siguientes:

- Circuitos de la unidad aritmética y lógica o ALU: Es la parte donde se realizan los cálculos y las decisiones lógicas para controlar el autómata.
- Circuitos de la unidad de control (UC) o Decodificador de instrucciones: Decodifica las instrucciones leídas en memoria y se generan las señales de control.
- Acumulador: Es la encargada de almacenar el resultado de la última operación realizada por el ALU.

- Flags: o indicadores de resultado, que pueden ser consultados por el programa.
- Contador de programa: Encargada de la lectura de las instrucciones de usuario.
- Bus (interno): No son circuitos en sí, sino zonas conductoras en paralelo que transmiten datos, direcciones, instrucciones y señales de control entre las diferentes partes del Controlador Lógico Programable.

Memoria monitor del sistema

Es una memoria de tipo ROM, Lectura y escritura en las interfaces de E/S. operativo del autómata contiene las siguientes rutinas, incluidas por el fabricante.

- Inicialización tras puesta en tensión o reset.
- Rutinas de test y de respuesta a error de funcionamiento.
- Intercambio de información con unidades exteriores.
- Lectura y escritura en las interfaces de E/S.
- Funciones básicas de la CPU.

En la memoria ROM del sistema, el fabricante ha grabado una serie de programas ejecutivos, software del sistema y es a estos programas a los que accederá el procesador para realizar las funciones.

El software del sistema de cualquier Controlador Lógico Programable consta de una serie de funciones básicas que realiza en determinados tiempos de cada ciclo.

En general cada Controlador Lógico Programable contiene y realiza las siguientes funciones:

- Vigilar que el tiempo de ejecución del programa de usuario, que no exceda de un determinado tiempo máximo. A esta función se le denomina Watchdog.

- Ejecutar el Programa del usuario Crear una imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.
- Renovar el estado de las salidas en función de la imagen de las mismas, obtenida al final del ciclo de ejecución del programa usuario.
- Cheque del sistema.

3.1.2.4.Memorias

La memoria es el almacén donde el Controlador Lógico Programable guarda todo cuanto necesita para ejecutar la tarea de control

- Datos Del proceso.
- Señales de planta, entradas y salidas.
- Variables internas, de bit y de palabra.
- Datos alfanuméricos y constantes.
- Datos de control.
- Instrucciones de usuario (programa).
- Configuración Controlador Lógico Programable (modo de funcionamiento, número de E/S conectadas, ...).

Existen varios tipos de memorias:

- RAM. Memoria de lectura y escritura.
- ROM. Memoria de solo lectura, no reprogramable.
- EPROM. Memoria de solo lectura, reprogramables con borrado por ultravioletas.
- EEPROM. Memoria de solo lectura, alterables por medios eléctricos.

La memoria RAM se utiliza principalmente como memoria interna, y únicamente como memoria de programa en el caso de que pueda asegurarse el mantenimiento de los datos con una batería exterior.

La memoria ROM se utiliza para almacenar el programa monitor del sistema como hemos visto en el apartado dedicado a la CPU.

Las memorias EPROM se utilizan para almacenar el programa de usuario, una vez que ha sido convenientemente depurada.

Las memorias EEPROM se emplean principalmente para almacenar programas, aunque en la actualidad es cada vez más frecuente el uso de combinaciones RAM + EEPROM (NOVRAM), utilizando estas últimas como memorias de seguridad que salvan el contenido de las RAM.

Una vez reanudada la alimentación, el contenido de la EEPROM se vuelca sobre la RAM. Las soluciones de este tipo están sustituyendo a las clásicas RAM + batería puesto que presentan muchos menos problemas.

Memoria interna

En un Controlador Lógico Programable, la memoria interna es aquella que almacena el estado de las variables que maneja, entradas, salidas, contadores, relees internos, señales de estado, etc. Esta memoria interna se encuentra dividida en varias áreas, cada una de ellas con un cometido y características distintas.

La clasificación de la memoria interna no se realiza atendiendo a sus características de lectura y escritura, sino por el tipo de variables que almacena y el número de bits que ocupa la variable. Así, la memoria interna del Controlador Lógico Programable queda clasificada en las siguientes áreas.

Área de imágenes de entradas/salidas y Área interna (IR).

En esta área de memoria se encuentran:

- Los canales (registros) asociados a los terminales externos (entradas y salidas).
- Los relees internos (no correspondidos con el terminal externo), gestionados como relees de E/S.
- Los relees E/S no usados pueden usarse como IR.

- No retienen estado frente a la falta de alimentación o cambio de modo de operación.

Área especial (SR). Son relees de señalización de funciones particulares como:

- Servicio (siempre ON, OFF)
- Diagnósticos (señalización o anomalías)
- Temporizaciones (relojes a varias frecuencias)
- Calculo
- Comunicaciones.
- Accesible en forma de bit o de canal.
- No conservan su estado en caso de fallo de alimentación o cambio de modo.

Área auxiliar (AR).

- Contienen bits de control e información de recursos de PLC como: puertos periféricos, casetes de memoria. Se dividen en dos bloques: Señalización: Errores de configuración, datos del sistema. Memorización y gestión de datos.
- Es un área de retención.
- Accesible en forma de bit o de canal.
- No conservan su estado en caso de fallo de alimentación o cambio de modo.

Área de enlace (LR).

- Dedicados al intercambio de información entre PLC's.
- Si no se utilizan como LR pueden usarse como IR.
- Accesible en forma de bit o canal.
- No conservan su estado en caso de fallo de alimentación o cambio de modo.

Área de retención (HR).

- Mantienen su estado ante fallos de alimentación o cambio de modo de PLC.
- Son gestionados como los IR y direccionables como bit o como canal.

Área de temporizadores y contadores (TIM/CNT).

- Es el área de memoria que simula el funcionamiento de estos dispositivos.
- Son usados por el PLC para programar retardos y cuentas.

Área de datos (DM).

- Se trata de memoria de 16 bits (palabra).
- Utilizable para gestión de valores numéricos.
- Mantiene su estado ante cambios de modos de trabajo o fallo de alimentación.
- Direccionables como Canal (palabra).
- Esta área suele contener los parámetros de configuración del PLC (setup).

Las variables contenidas en la memoria interna, pueden ser consultadas y modificadas continuamente por el programa, cualquier número de veces. Esta actualización continua de los datos obliga a construir la memoria con dispositivos RAM.

Memoria de programa

La memoria de programa, normalmente externa y enchufable a la CPU mediante casete de memoria, almacena el programa escrito por el usuario para su aplicación.

Cada instrucción del usuario ocupa un paso o dirección del programa.

Las memorias de programa o memorias de usuario son siempre de tipo permanente RAM + batería o EPROM / EEPROM. Por lo general la mayoría de los fabricantes de autómatas ofrecen la posibilidad de utilizar memorias RAM con batería para la fase de desarrollo y depuración de los programas, y de pasar estos a memorias no volátiles EPROM o EEPROM una vez finalizada esta fase.

La ejecución del programa en el módulo es siempre prioritaria, de forma que si se da tensión al autómata con un módulo conectado, la CPU ejecuta su programa y no el contenido en memoria RAM interna.

3.1.2.5.Interfaces

En el control de un proceso automatizado, es imprescindible un dialogo entre operador máquina junto con una comunicación entre la máquina y el Controlador Lógico Programable, estas comunicaciones se establecerán por medio del conjunto de entradas y salidas del citado elemento.

Todas las señales provenientes del campo son informadas a la CPU, luego de ser tomadas por los captosres de entradas, y a su vez, las órdenes generadas por la CPU son comunicadas a los elementos del proceso bajo control por medio de las interfaces de salida.

Los Controlador Lógico Programable son capaces de manejar tensiones y corrientes de nivel industrial, gracias a que disponen un bloque de circuitos de interfaz de E/S muy potente, que les permite conectarse directamente con los sensores y accionamientos del proceso.

En los controladores más sencillos, las interfaces de entrada se encargan de convertir la tensión o la corriente que reciben de los sensores, límites de carrera, pulsadores, llaves, etc., en niveles apropiados para la operación de la CPU. De la misma manera las interfaces de salida permiten partiendo de las señales de baja tensión originadas en la CPU, comandar contactores, solenoides de válvulas, arrancadores de motores, valiéndose de diacs, triacs, relés etc.

Las señales digitales o discretas como los interruptores, son simplemente una señal de 1 ó 0, Verdadero o Falso, respectivamente. Los interruptores son ejemplos de dispositivos que proporcionan una señal discreta, que son enviadas usando la tensión o la intensidad, donde un rango específico corresponderá al On y otro rango al Off. Un PLC puede utilizar 24V de corriente continua en la E/S donde valores superiores a 22V representan un On, y valores inferiores a 2V representan Off. Inicialmente los PLC solo tenían E/S discretas.

A medida que la complejidad de los PLC aumenta, es necesario contar con otro tipo de interfaces que puedan interpretar señales analógicas provenientes del proceso y emitirlas como salidas.

Las señales analógicas son como controles de volúmenes, con un rango de valores entre 0 y el tope de escala. Esto es normalmente interpretado con valores

enteros por el PLC, con varios rangos de precisión dependiendo del dispositivo o del número de bits disponibles para almacenar los datos. Presión, temperatura, flujo, y peso son normalmente representados por señales analógicas. Las señales analógicas pueden usar tensión o intensidad con una magnitud proporcional al valor de la señal que procesamos.

A medida que los requerimientos de control se hacen más complicados, aparecen los sistemas inteligentes, periféricos cuentan con un microprocesador propio, que descargan en parte el trabajo de la CPU, para hacer más rápida la ejecución del programa del usuario. Estas interfaces inteligentes pueden manipular datos, resolver ecuaciones aritméticas, comparaciones, conteos de alta velocidad.

De entre todos los tipos de interfaces que existen, las interfaces específicas permiten la conexión con elementos muy concretos del proceso de automatización. Se pueden distinguir entre ellas tres grupos bien diferenciados:

- Entradas / salidas especiales.
- Entradas / salidas inteligentes
- Procesadores periféricos inteligentes.

Las interfaces especiales del primer grupo se caracterizan por no influir en las variables de estado del proceso de automatización. Únicamente se encargan de adecuar las E/S, para que puedan ser inteligibles por la CPU, si son entradas, o para que puedan ser interpretadas correctamente por actuadores (motores, cilindros, etc.), en el caso de las salidas.

Las del segundo grupo admiten múltiples modos de configuración, por medio de unas combinaciones binarias situadas en la misma tarjeta. De esta forma se descarga de trabajo a la unidad central, con las ventajas que conlleva.

Los procesadores periféricos inteligentes, son módulos que incluyen su propio procesador, memorias y puntos auxiliares de entrada / salida. Estos procesadores contienen en origen un programa especializado en la ejecución de una tarea concreta, a la que le basta conocer los puntos de consigna y los parámetros de aplicación para ejecutar, de forma autónoma e independiente de la CPU principal, el programa de control.

Entradas - Salidas

La sección de entradas mediante el interfaz, adapta y codifica de forma comprensible para la CPU las señales procedentes de los dispositivos de entrada o captadores.

Hay dos tipos de entradas:

- Entradas digitales
- Entradas analógicas

La sección de salida también mediante interfaz trabaja de forma inversa a las entradas, es decir, decodifica las señales procedentes de la CPU, y las amplifica y manda con ellas los dispositivos de salida o actuadores como lámparas, relees... aquí también existen unos interfaces de adaptación a las salidas de protección de circuitos internos.

Hay dos tipos de salidas:

- Salidas digitales
- Salidas analógicas

Entradas y Salidas Discretas

Estas interfaces tiene la simple función de informar a la CPU, de la presencia o ausencia de señal, tensión o corriente, en un circuito, apertura o cierre de un contacto, pulsador, límite de carrera, etc. En el caso de las salidas estas conectan o desconectan al circuito de actuación de un solenoide, contactor, lámpara, etc.

Las interfaces discretas abarcan un rango muy amplio de opciones de operación. Un contacto externo al controlador puede estar conectado a distintos voltajes, según la máquina o proceso lo mismo para otro tipo de captor.

Existen entonces interfaces para corriente alterna, corriente continua y a su vez para distintos niveles y tipos de tensiones que van desde los cinco voltios hasta niveles industriales.

Las interfaces de entrada-salida suelen estar construidas de forma de módulos que se alojan en bases de montaje, controladores modulares, o bien formando parte del controlador, compactos.

Tanto las entradas como las salidas pueden contener un borne común, para varias de ellas o bien estar dispuestas en forma individual aisladas entre sí.

Los módulos de entrada digitales permiten conectar al Controlador Lógico Programable de tipo todo o nada como finales de carrera pulsadores, llaves, etc.

Los módulos de entrada digitales trabajan con señales de tensión, por ejemplo cuando por una vía llegan 24 voltios se interpreta como un "1" y cuando llegan cero voltios se interpreta como un "0".

El proceso de adquisición de la señal digital consta de varias etapas.

- Protección contra sobre tensiones
- Filtrado
- Puesta en forma de la onda
- Aislamiento galvánico o por opto acoplador.

Las entradas digitales pueden ser discretas o de palabras:

Entradas Discretas

Dentro de cada interface de entrada discreta, existe un elemento rectificador y un acondicionador de señal que elimina los ruidos de líneas y rebates de contactos.

Luego un tercer elemento detecta el umbral de tensión de activación y finalmente a través de una opto aislación se conecta a la lógica de la CPU. Esta aislación es para asegurar el funcionamiento confiable del controlador.

El último bloque de una entrada es el que comunica a la lógica del sistema un uno o un cero según el nivel de tensión de entrada. Esto es siempre complementado por un indicador de nivel del estado de entrada constituido generalmente por un LED.

El estado activado – desactivado de cada entrada se guarda en tablas de memorias, memorias imagen o registro de imagen, para que una vez por cada

barrido del programa del usuario informen su estado a la CPU y luego se vuelven a actualizar.

Estas interfaces tienen la simple función de informar a la CPU del estado de presencia o ausencia de tensión en un circuito (cierre o apertura de un Contactor, pulsador, etc.). Las interfaces discretas abarcan un amplio rango de opciones, un contacto externo puede estar conectado a distintos voltajes, según la máquina o el proceso. Existen interfaces para corriente continua o alterna, y a su vez para distintos niveles de tensiones.

Las interfaces están construidas de forma de módulos que se alojan en bases de montaje, estructura semimodular o modular, o formando parte del conjunto estructura compacta.

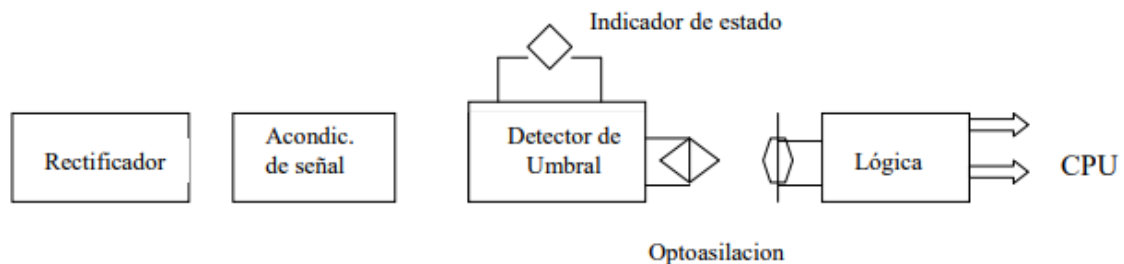


Figura 2.18 Estructura de un circuito de entrada discreta indicando las etapas desde la entrada digital hasta el CPU.

Salidas Discretas

Un módulo de salida digital permite al Consolador Lógico Programable actuar sobre los preaccionadores y accionadores que admitan ordenes de tipo todo o nada.

El valor binario de las salidas digitales se convierte en la apertura o cierre de un relee interno del autómatas en el caso de módulos de salidas a relé.

En los módulos estáticos (bornero), los elementos que conmutan son los componentes electrónicos como transistores o triacs, y en los módulos electromecánicos son contactos de relees internos al módulo.

Los módulos de salidas estáticas al suministrar tensión, solo pueden actuar sobre elementos que trabajan todos a la misma tensión, en cambio los módulos de salida electromecánicos, al ser libres de tensión, pueden actuar sobre elementos que trabajen a tensiones distintas.

El proceso de envío de la señal digital consta de varias etapas:

- Puesta en forma.
- Aislamiento Circuito de mando (relé interno).
- Protección electrónica.
- Tratamiento cortocircuitos.

Las interfaces de salida discretas son similares, la señal de activación originada por la CPU, es pasada por una opto aislación, seguida por un circuito que se encarga de disparar el elemento final de salida, relés, triacs, bobina, transistor de potencia, etc., existe además un elemento de protección contra sobrecargas o corto circuitos.

La detección del nivel de umbral de entrada causa una demora que varía según el fabricante y se estima en un valor promedio de 10 milisegundos, esta demora no es igual cuando se trata de conexión que de desconexión, siendo las salidas discretas que trabajan con corriente continua son más veloces que las de corriente alterna.

También, como en las entradas, el estado activado – desactivado de cada salida se guarda en tablas de memorias imagen o en registros de imagen, para que una vez por cada barrido del programa del usuario informen su estado a la CPU y luego se vuelven a actualizar.

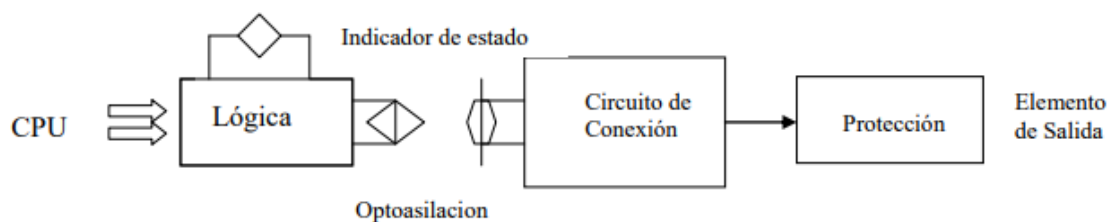


Figura 2.19 Estructura de un circuito de salida digital, indicando sus etapas desde el CPU hasta el elemento de salida.

Entradas de Palabras

Las interfaces de entradas de palabras permiten conectar elementos cuyas señales son palabras formadas por múltiples bits en paralelos, como por ejemplo llaves selectoras rotativas binarias.

Las palabras pueden estar formadas por cuatro u ocho canales de entrada, cada canal permite conectar varias llaves del tipo binario, este tipo de entrada cuenta con un borne para activarlas de forma multiplexada de uno en uno en forma secuencial.

Las entradas filtradas y luego en forma sincronizada se almacena en una memoria intermedia y permanecen allí hasta ser leídas, cuando la actualización de cada canal no es sincrónico con el barrido de la CPU, existe un sistema de protección para que el canal no se actualice en el momento en que esta leído.

Salidas de Palabras

Las salidas de este tipo activan grupos de 8 o 16 bits, que forman palabras binarias. Se usan para manejar elementos como visualizador de siete segmentos y otros elementos del proceso capaz de capaz este tipo de señales.

Generalmente cuentan con una única bornera de 8/16 bornes y de una señal para sincronizar la lectura de varios canales por esa misma bornera. Las salidas son opto aisladas y poseen un sistema para evitar la actualización de los canales mientras se están siendo leídos

Entradas analógicas

Los módulos de entrada analógicas permiten que los Controlador Lógico Programable trabajen con accionadores de mando analógico y lean señales de tipo analógico como pueden ser la temperatura, la presión, el caudal, tensión o intensidad, etc.

Los módulos de entradas analógicas convierten una magnitud analógica en un número que se deposita en una variable interna del Controlador Lógico

Programable. Lo que realiza es una conversión A/D, puesto que el autómata solo trabajar con señales digitales. Esta conversión se realiza con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo).

El proceso de adquisición de la señal analógica consta de varias etapas:

- Filtrado
- Conversión A/D
- Memoria interna

Un módulo clásico de entrada analógica puede tener, por ejemplo cuatro u ocho canales de entrada multiplexado. Poseen normalmente etapas en sus circuitos con filtros y limitaciones de señal. La etapa limitadora previene la llegada al conversor analógico/digital de señales de valor excesivo o de polaridad incorrecta, las señales pueden ser además opto aisladas tanto en la entrada como en el multiplexado.

Luego de pasar por las etapas de filtrado y limitación, la señal analógica es transmitida al conversor analógico digital desde donde la señal digital equivalente pasa por una memoria intermedia y luego a la memoria de estados de entrada y salidas del controlador, dentro de esta etapa también se encuentran circuitos de sincronismos para seleccionar el canal que debe ser leído en forma secuencial y transportar el valor hasta la memoria intermedia.

También posee circuitos de inhibición para evitar la lectura por la CPU simultáneamente de valores en la memoria intermedia, de la misma manera que en los módulos de entrada.

El tiempo de lectura y actualización de los estados de entrada analógica está determinado por el módulo en sí y es independiente del tiempo de barrido de la CPU. De otro modo, el tiempo no depende de cuantas veces lee la CPU en estado de la memoria intermedia, sino de la mayor o menor velocidad del ciclo del conversor analógico/digital.

Luego del proceso de lectura, los datos obtenidos se transfieren a posiciones de memoria, desde donde el programa escrito por el usuario toma los valores para realizar las operaciones.

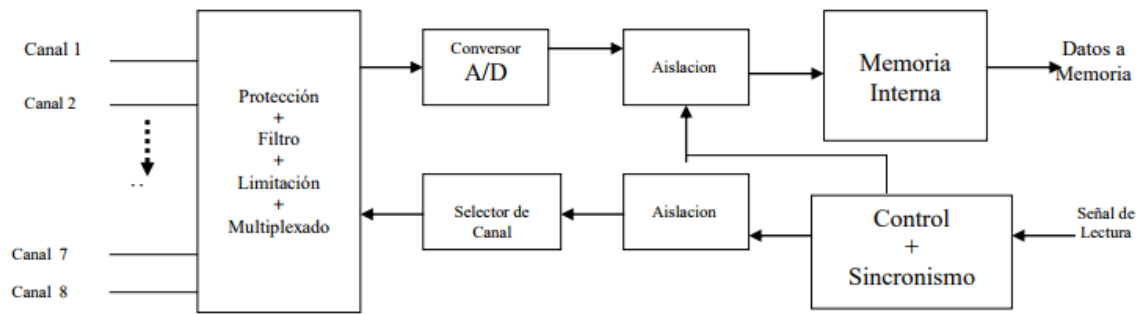


Figura 2.20 Estructura por etapas de los circuitos para entrada analógica.

Salidas analógicas

Los módulos de salida analógica permiten que el valor de una variable numérica interna del autómatas se convierta en tensión o intensidad.

Lo que realiza es una conversión D/A, puesto que el Controlador Lógico Programable solo trabaja con señales digitales. Esta conversión se realiza con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo).

Esta tensión o intensidad puede servir de referencia de mando para actuadores que admitan mando analógico como pueden ser los variadores de velocidad, las etapas de los tiristores de los hornos, reguladores de temperatura... permitiendo al Controlador Lógico Programable realizar funciones de regulación y control de procesos continuos.

El proceso de envío de la señal analógica consta de varias etapas:

- Aislamiento galvánico
- Conversión D/A
- Circuitos de amplificación y adaptación
- Protección electrónica de la salida

Como hemos visto las señales analógicas sufren un gran proceso de adaptación tanto en los módulos de entrada como en los módulos de salida. Las funciones de conversión A/D y D/A que realiza son esenciales. Por ello los módulos de E/S analógicos se les consideran módulos de E/S especiales.

Los tamaños de las palabras son de 10 bits o menores, esa longitud de palabra da una resolución de una parte en mil veinticuatro ($1/1024$) que corresponde aproximadamente 0,01 voltios si se trabaja entre 0 y 10 voltios.

Los módulos de este tipo pueden manejar entre cuatro u ocho salidas, tiene circuitos de aislación antes de entrar al conversor digital/analógico, control de sincronismo y control para evitar choques entre la lectura y escritura de cada una de las salidas de los canales.

3.1.2.6. Módulo de Funciones Especiales

Todas las interfaces o adaptadores descritos hasta ahora tienen la tarea de convertir señales de entrada en valores aceptables para la CPU, o convertir las señales entregadas por la CPU en valores convenientes para los actuadores.

Cuando los PLC deben controlar procesos o máquinas que requieren tareas más complejas, como por ejemplo, resolución de ecuaciones que requieren aritmética avanzada, emisión de informes en códigos ASCII, control de velocidades superiores al barrido del equipo, repuestas a señales que no pueden aceptar demoras, control de lazos PID, estas tipo de tareas pueden resultar limitaciones que estén dadas por la falta de capacidad de los controladores para atender en un tiempo razonable esas operaciones sin dejar de lado la resolución de la lógica de contacto.

Para ello se diseñaron módulos de entrada/salida con concepto de modulo inteligente de funciones especiales, estos tiene la capacidad propia para el procesamiento de datos y no influyen en el tiempo de barrido del contador, por contar con su propio microprocesador y un barrido asincrónico con respecto a la CPU, pero con la capacidad de tomar, modificar y escribir datos en la memoria.

Módulo de Entrada de Pulsos de Alta Velocidad

Permite conectar al sistema dispositivos que producen trenes de pulsos demasiado rápidos para que el barrido del controlador, pueda reaccionar o efectuar conteos, ejemplo, caudalímetro, turbinas, tacómetros, etc.

Este tipo de módulos tiene entradas/salidas, por las que ingresan las señales de frecuencias hasta 100 KHz., y las salidas pueden adoptar resultados de conexión/desconexión de acuerdo al programa que el usuario carga en la memoria del módulo.

El programa consiste fundamentalmente comparar entre los valores reales de conteo, con los que el usuario prefija, o pueden ser tomados de la memoria principal del PLC, cuando se alcanzan los valores prefijados, se activan las salidas del módulo y/o se actualizan los estados de la memoria del PLC.

Cuando se usa un caudalímetro que envía pulsos, el módulo puede efectuar conteos de acuerdo a una unidad de tiempo, pulsos/segundos.

Módulo de Control de Ejes

Este módulo tiene generalmente la función de controlar la posición punto a punto de servomotores en lazos cerrados, tienen la posibilidad de manejar el posicionamiento de varios ejes a la vez.

La tarea principal del módulo es el cómputo de velocidad y posición independientemente del barrido del PLC. Para ello el módulo, cuenta con su propia CPU y se programa con el lenguaje de alto nivel por medio de una PC o una terminal sin inteligencia.

Además estos módulos pueden almacenar distintos perfiles de funcionamiento en tablas que son consultadas desde el programa del usuario.

Este tipo de módulo cuenta con puertos de comunicación que pueden usarse para emitir reportes a una impresora o algún periférico en serie, centro de mecanización, máquinas, herramientas, robots de soldadura, etc.

Módulo de Entrada de Termocuplas

Proveen de alta precisión en las lecturas de termocuplas, usando resoluciones de hasta catorce bits, la señal que se le envía al PLC puede ser expresada en grados Celsius, grados Fahrenheit, o en milivoltios.

La calibración del módulo, ajuste de cero y rango se realizan de manera similar a los métodos de calibración de transmisores, estos ajustes dan la posibilidad de lograr una alta resolución en bajas temperaturas manteniendo el rango al máximo.

El módulo permite la conexión de varias termocuplas, que constituyen canales de entrada que se interpretan secuencialmente mediante un barrido propio del módulo. El tiempo aproximado para ocho canales es de 40 milisegundos, los canales que no se usen pueden ser deshabilitados para aumentar la velocidad de barrido.

Módulo de Entrada de Termo resistencias

Mediante la lectura de un valor de resistencia y su posterior conversión, este módulo es capaz de enviar señales a la CPU del PLC en varios formatos, grados Celsius, grados Fahrenheit, Ohmios o valor numérico. El módulo posee una inteligencia necesaria para convertir el valor de resistencia leído en formato útil para el usuario sin el empleo de tablas de conversiones.

Otra de las ventajas de este tipo de módulo son las compensaciones de la resistencia de cables y capacidad de detectar circuitos abiertos.

Módulo BASIC Programable

Cuando se necesitan cálculos complejos, estadística, etc. se puede utilizar este tipo de módulo, que no es más que una pequeña computadora con su propia CPU, y memorias que acepta programas escritos en lenguaje BASIC. El módulo posee una comunicación directa con la CPU del PLC y otros puertos para poder conectar periféricos tales como terminales sin inteligencia, lectores de códigos de barra, impresoras, etc.

Con el uso del Basic se pueden hacer cálculos para resolver lazos de control PID, cuando el PLC no incluya como capacidad propia, el programa se almacena en RAM, soportadas por baterías, pero con la posibilidad de transferir en forma automática, datos a memorias que permanezcan inalterables ante falta de energía.

Otra de las aplicaciones es la de mantener en la memoria del módulo, una cantidad de recetas de distintos productos a elaborar, las cuales se pueden descargar a

pedido del operador en el momento adecuado, modificando las posiciones de memoria requeridas por el PLC.

3.1.3. LENGUAJE LADDER

De acuerdo a lo que se menciona en Wikipedia (2017), el lenguaje Ladder, diagrama de contactos, o diagrama en escalera, es un lenguaje de programación gráfico muy popular dentro de los autómatas programables debido a que está basado en los esquemas eléctricos de control clásicos. De este modo, con los conocimientos que todo técnico eléctrico posee, es muy fácil adaptarse a la programación en este tipo de lenguaje.

Ladder es uno de los varios lenguajes de programación para los controladores lógicos programables (PLCs) estandarizados con la norma IEC 61131-3.

Para programar un autómata con Ladder, además de estar familiarizado con las reglas de los circuitos de conmutación, (también denominada Lógica de Contactos), es necesario conocer cada uno de los elementos de que consta este lenguaje. A continuación, se describen de modo general los más comunes.

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa; esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y, usada junto con la bobina RESET, dan una enorme potencia en la programación.
	Bobina JUMP	Permite saltarse instrucciones del programa e ir directamente a la etiqueta que se desee. Sirve para realizar subprogramas.

Figura 2.21 Elementos básicos en Ladder. Figura cortesía de Wikipedia 2017.

Se suele indicar mediante los caracteres B ó M y tienen tanto bobinas como contactos asociados a las mismas de los tipos vistos en la figura 3. Su número de identificación suele oscilar, en general, entre 0 y 255. Su utilidad fundamental es la de almacenar información intermedia para simplificar esquemas y programación.

Los bits de sistema son contactos que el propio autómata activa cuando conviene o cuando se dan unas circunstancias determinadas. Existe una gran variedad,

siendo los más importantes los de arranque y los de reloj, que permiten que empiece la ejecución desde un sitio en concreto y formar una base de tiempos respectivamente. Su nomenclatura es muy diversa, dependiendo siempre del tipo de autómatas y fabricante.

En forma práctica, los contactos pueden representar, de acuerdo a la programación que se realice, la entrada de información, por ejemplo el activamiento de un sensor, o el activamiento de una memoria interna, comúnmente denominada M.

Las Bobinas, representan a esas memorias M así como las salidas al exterior, activan o desactivan elementos como puede ser una luz, un motor eléctrico, un solenoide de una válvula neumática, etc.

3.2. ELABORACIÓN DE LA HIPÓTESIS

De acuerdo con la formulación del problema y el objetivo planteado, la hipótesis de investigación concreta a esta situación problemática, queda enunciada bajo la siguiente proposición:

La información sobre los controladores existentes permitirá diseñar un programador lógico programable usando microcontrolador ATMEGA y lenguaje LADDER para aplicaciones de laboratorio.

CAPÍTULO 3: METODOLOGÍA

6.1. TIPO DE INVESTIGACIÓN, MÉTODO DE INVESTIGACIÓN

6.1.1. Tipo de investigación

Tomando en cuenta que se busca formular un diseño con el objetivo de plantear una solución a la realidad problemática, nuestra investigación será **aplicada**.

6.1.2. Método de investigación

El método que seguiremos, es el método de ingeniería que conlleva el usar los siguientes pasos:

- Identificación del problema
- Recolección de la información necesaria
- Búsqueda de soluciones creativas
- Pasar de la idea principal al diseño preliminar
- Evaluación y selección de la solución
- Preparación de reportes, planos y especificaciones
- Implementación del diseño.

CAPÍTULO 4: DISEÑO DEL SISTEMA

4.1. DESARROLLO DE HARDWARE

Para el desarrollo del hardware del programador lógico programable, tendremos en cuenta la cantidad de entradas y salidas que se podría necesitar para un proyecto de laboratorio.

Los PLC compactos típicamente vienen con pocas entradas y salidas, lo ideal para aplicaciones pequeñas, por ejemplo el gobernar el arranque de un motor; en cambio los PLC modulares, utilizan diversos módulos para ampliar la cantidad de entradas y salidas, de acuerdo a lo que requiere un proyecto o incluso una futura ampliación.

Para nuestro proyecto, se va a diseñar un PLC con las siguientes entradas y salidas:

ÍTEM	DESCRIPCIÓN	CANTIDAD
1	Entradas digitales tipo contacto, con alimentación disponible de 24VDC	8
2	Salidas digitales tipo relé, con capacidad para soportar alimentación alterna o continua	8
3	Entradas analógicas de -10V a 10V, con resolución de 10 bits	4
4	Salidas analógicas de 0 a 10V, con etapa de potencia incorporada	4
5	Comunicación vía USB para la programación del PLC	1
6	Fuente de alimentación incorporada de entrada 24VDC	1

Tabla 4.1. *Tabla de especificaciones mínimas del PLC compacto a diseñar.*

Con ello podremos apreciar que se va a diseñar un PLC compacto, con la capacidad de poder realizar experimentos de laboratorio, ya que se podría integrar a los módulos Leybold que se tienen.

A continuación se darán las pautas de diseño de cada parte del sistema en conjunto.

4.2. SELECCIÓN DE LOS CIRCUITOS PARA LA FUENTE DE ALIMENTACIÓN

El voltaje de alimentación alterna que se utiliza comúnmente a nivel industrial es 24 voltios de corriente directa, por ello, se va a utilizar una fuente cuya entrada, justamente, utilice este voltaje, además de ello, las entradas y salidas digitales, también utilizan 24 voltios continuos para su funcionamiento.

Ahora bien, los dispositivos electrónicos de procesamiento utilizan 5 voltios continuos, por ello, se utilizarán circuitos reguladores de tensión para reducir el voltaje de 24 voltios continuos a 5 voltios continuos.

Adicionalmente, se utilizarían micro relés, que típicamente utilizan una bobina de 12 voltios continuos, es por ello, que adicional a la regulación a 5 voltios continuos, se utilizará un regulador a 12 voltios continuos.

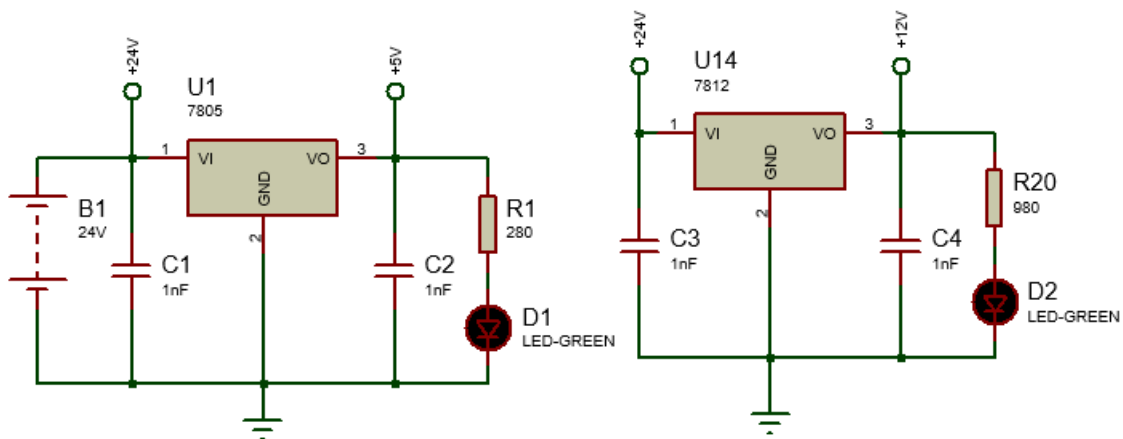


Figura 4.1. Circuito regulador para la fuente del controlador lógico programable.

4.2.1. Reguladores de voltaje

Con lo expuesto en la sección anterior, el circuito propuesto queda similar a lo que se indica en la figura 4.1, como se puede apreciar se está utilizando los reguladores

7805 (U1 en la figura 4.1) y 7812 (U14 en la figura 4.1) como reguladores de voltaje de 24 voltios a 5 voltios continuos y a 12 voltios continuos respectivamente.

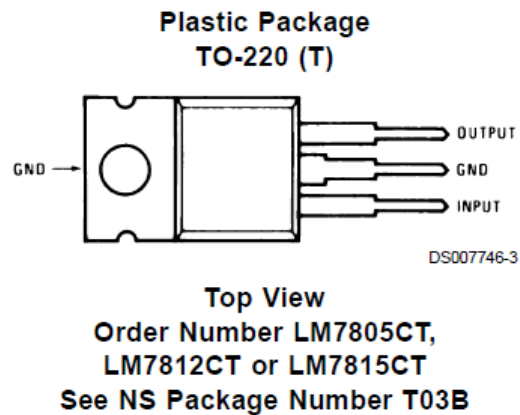


Figura 4.2. Carcasa de los reguladores 7805 y 7812, cortesía de National Semiconductor, hoja de datos en: http://pdf.datasheetcatalog.com/datasheets/150/44435_DS.pdf.

Tome en consideración que la corriente de salida de cada regulador es de 1 amperio, lo suficiente para alimentar los circuitos.

4.3. SELECCIÓN DE LA UNIDAD DE CONTROL DE PROCESOS

La unidad de control de procesos lo conforma un microcontrolador, que estará a cargo de almacenar el programa que permita controlar un determinado proceso.

De acuerdo a lo indicado en la realidad problemática, el microcontrolador seleccionado será un Atmel, específicamente un circuito Arduino.

Como ya se sabe, existen diversos tipos de Arduinos, los más destacados son el Arduino UNO, el Arduino NANO y el Arduino Mega.

Para la selección del Arduino más adecuado para la aplicación, se tomará en cuenta que se necesita la cantidad de entradas y salidas vistas en la tabla 4.1.

El Arduino Mega es probablemente el microcontrolador con mayores cualidades de la familia Arduino ya que posee, por ejemplo: 54 pines digitales que pueden funcionar como entrada/salida (14 de las cuales pueden ser utilizadas como salidas PWM); 16 pines de entradas análogas, un cristal oscilador de 16 MHz, una conexión USB para comunicarse con el computador, un boton de reset y una entrada para la alimentación de la placa.

Una de las últimas versiones del Arduino Mega 2560 adicionalmente a todas las características de su antecesor, utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI (circuito de conversión USB-serie). Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

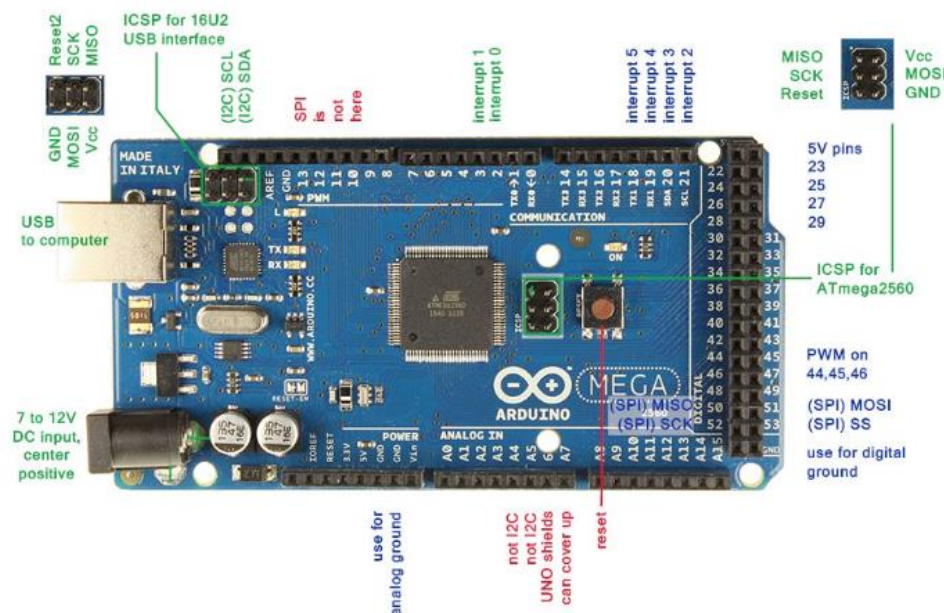


Figura 4.3. Disposición de pines del Arduino MEGA, imagen en:

<http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>

La comunicación entre la computadora y Arduino se produce a través del Puerto Serie. Posee un convertidor USB-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB.

4.3.1. Alimentación del Arduino MEGA

Arduino Mega puede ser alimentado mediante el puerto USB o con una fuente externa de poder. La alimentación es seleccionada de manera automática.

Cuando se trabaja con una fuente externa de poder se debe utilizar un convertidor AC/DC y regular dicho voltaje en el rango operativo de la placa. De igual manera se puede alimentar el micro mediante el uso de baterías. Preferiblemente el voltaje debe estar en el rango de los 7V hasta los 12V.

Arduino Mega posee algunos pines para la alimentación del circuito aparte del adaptador para la alimentación:

- VIN: A través de este pin es posible proporcionar alimentación a la placa.
- 5V: Podemos obtener un voltaje de 5V y una corriente de 40mA desde este pin.
- 3.3V: Podemos obtener un voltaje de 3.3V y una corriente de 50mA desde este pin.
- GND: El ground (0V) de la placa.

Dentro de las características más saltantes son las siguientes:

- Microcontrolador: ATmega2560.
- Voltaje de operación: 5VDC.
- Voltaje de entrada (recomendada): 7-12VDC.
- Voltaje de entrada (límites): 6-20VDC.
- Entradas y salidas digital: 54 (con capacidad para 15 salidas PWM).

- Entradas analógicas: 16.
- Corriente continua por entrada y salida: 20 mA.
- Corriente continua por el pin de 3.3V: 50 mA.
- Memoria Flash de 256 KB con 8 KB usados para bootloader.
- SRAM: 8 KB.
- EEPROM: 4 KB.
- Velocidad de 16 MHz.
- LED de prueba: pin 13.
- Longitud: 101.52mm.
- Ancho: 53.3mm.
- Peso: 37g.



Figura 4.4. Arduino MEGA, imagen en: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

A continuación se muestra la asignación de pin para el Atmega2560.

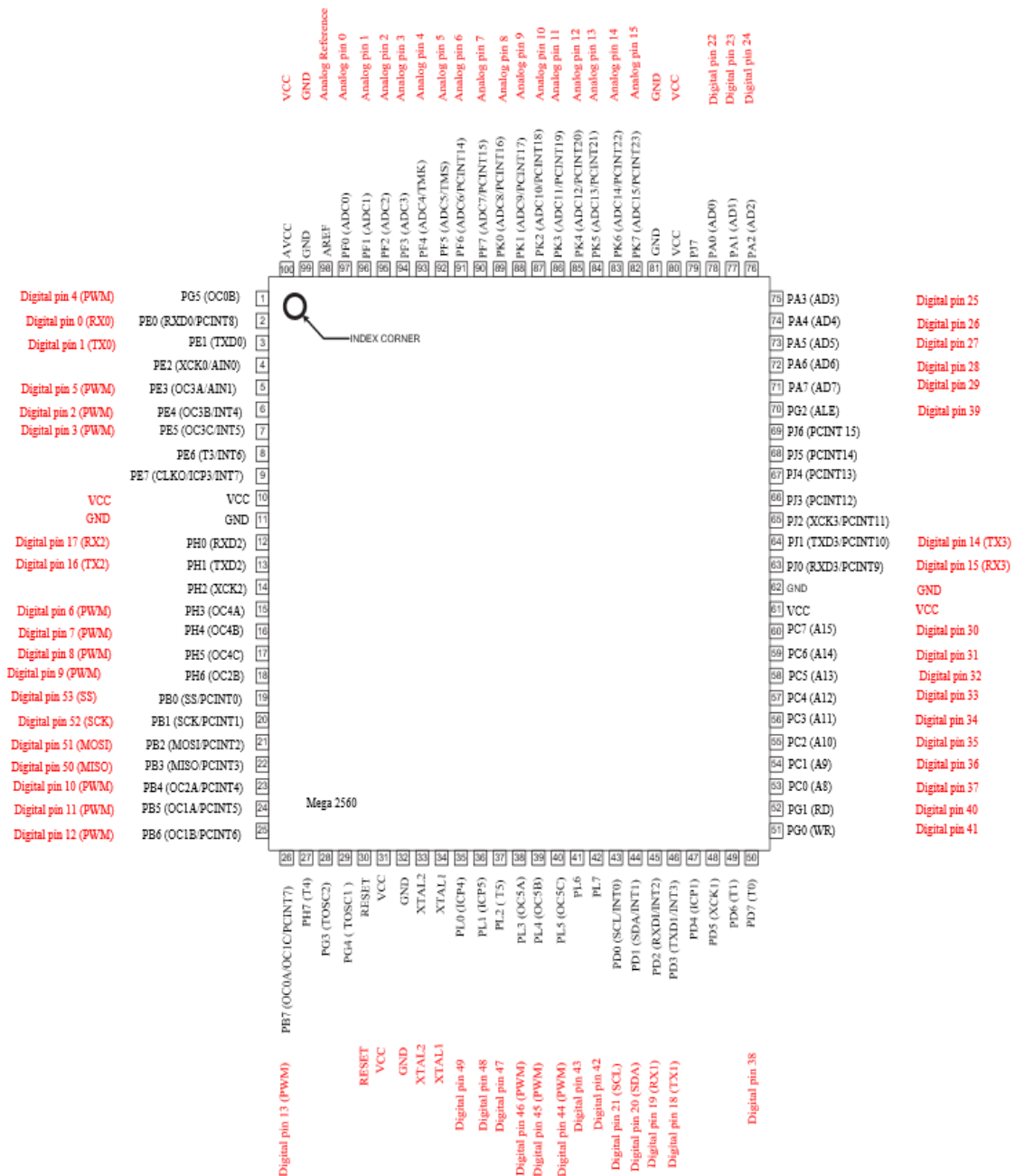


Figura 4.5. Disposición de pines del Atmega 2560, en: <https://www.arduino.cc/en/Hacking/PinMapping2560>

4.4. DISEÑO DE ENTRADA ANALÓGICA

Para el diseño propuesto se necesita de 8 entradas analógicas, en consecuencia se necesitará un convertidor de analógico a digital de 8 canales para manipular estas señales. Es así que uno de los requisitos para la elección del microcontrolador,

será que posea internamente un convertidor A/D, para así evitar el empleo de un convertidor A/D externo de varios canales o uno de canal simple agregando un multiplexor de 8 canales.

En ésta sección asumiremos que el microcontrolador a elegir poseerá un convertidor A/D de 8 canales y que el voltaje de entrada es de 0 a 5 voltios.

Una característica más es que la entrada analógica es de -10 voltios a +10 voltios, es por ello que ésta entrada no puede conectarse directamente al canal analógico del microcontrolador, es así que se necesita agregar una etapa de acondicionamiento de señal que permita convertir la entrada analógica de -10 a +10 voltios a una de 0 a 5 voltios, para que el convertidor A/D no se dañe.

4.4.1. Etapa de acondicionamiento de señal

Para el diseño de la etapa de acondicionamiento de señal se recurrirá a un circuito basado en resistencias como se muestra en la figura 4.6 y que en los siguientes párrafos se explicará su funcionamiento.

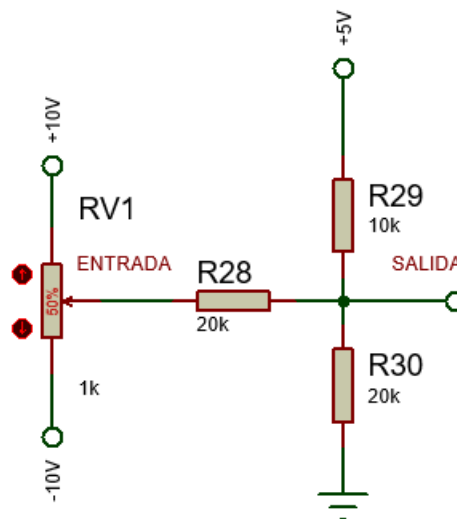


Figura 4.6. Circuito de entrada analógica acondicionada.

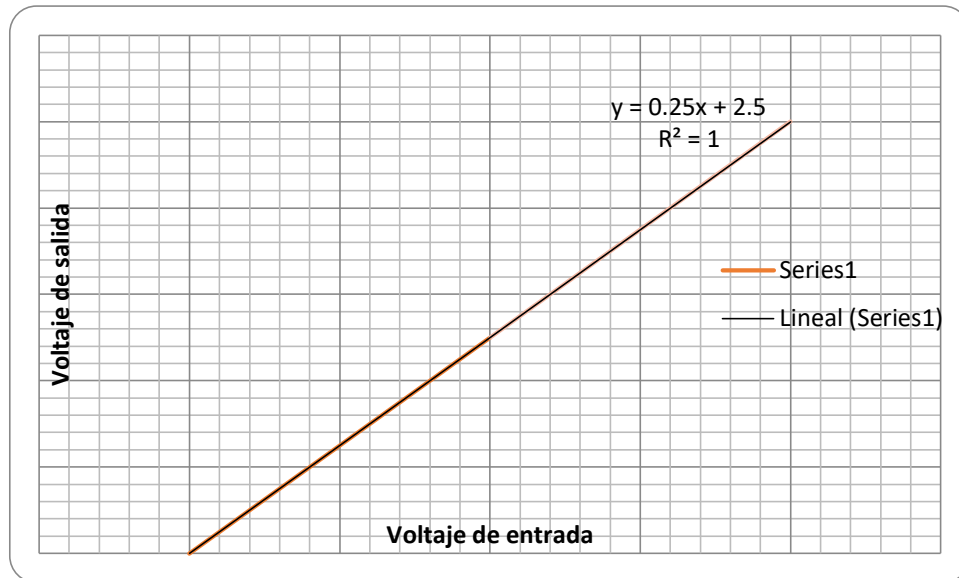


Figura 4.7. Curva de voltaje de salida vs. voltaje de entrada.

De acuerdo a lo visto en el circuito de la figura 4.6, podemos sacar algunas conclusiones, veamos, que sucede cuando en la entrada se coloca un voltaje equivalente a 0 voltios, lo que sucedería sería que la resistencia R_1 prácticamente se conectaría a tierra y por lo tanto estaría en paralelo con la resistencia R_2 , esto provocaría que existiría una resistencia equivalente igual R , ya que R_1 es igual a R_2 que es igual a $2R$, finalmente para encontrar el voltaje en la puerta analógica, se obtiene, por divisor de tensión, 2.5 voltios.

Si proseguimos colocando una serie de voltajes de entrada, aplicando reducción de circuitos y divisor de tensión obtendremos los voltajes de 0 a 5 voltios que son los ideales que utiliza en conversor A/D interno del microcontrolador.

Finalmente concluimos que el circuito es prácticamente lineal como lo expresa curva que se muestra en la figura 4.7.

4.5. DISEÑO DE LA SALIDA ANALÓGICA

De acuerdo a los requerimientos del diseño, se necesita de dos salidas analógicas de 0 a 10V, para esto se tienen algunas alternativas, una de ellas puede ser un convertidor de digital a analógico, que es un dispositivo externo al microcontrolador, el dispositivo clásico, por ejemplo el DAC0808, es de 8 bits de resolución y por lo tanto 8 bits de entrada para una salida lineal típica de 0 a +10V.

Este dispositivo necesita de fuentes de alimentación de 5 voltios, 10, voltios y otros voltajes, algo muy tedioso, debido a que se necesitaría diseñar circuitos de regulación de voltaje. Otro motivo por el cual no usar este dispositivo para nuestro diseño es que se necesitaría más de una puerta para manipular dicho conversor, finalmente concluimos que este dispositivo no sería el más adecuado.

Otra posibilidad es utilizar potenciómetros digitales, uno de los más utilizados es el DS1267, el impedimento principal es el costo, y que no se puede encontrar fácilmente.

Pero existe una solución más sencilla y que haría uso directamente del microcontrolador, nos referimos a utilizar dos salidas moduladoras por ancho de pulso (PWM).

4.5.1. Etapa de potencia

El microcontrolador tiene internamente dos PWM que se pueden configurar para obtener una salida analógica, para esto se utiliza un circuito RC, es decir una resistencia y un condensador. Según Barra (2011), para utilizar la salida PWM se recomienda una resistencia de $1k\Omega$ y un condensador de $10\mu F$. Con esto obtendremos una salida máxima de +5V es por ello, que para nuestro diseño colocaremos un amplificador de ganancia 2 con un amplificador TL082 y un transistor de potencia TIP41 para obtener ganancia en corriente.

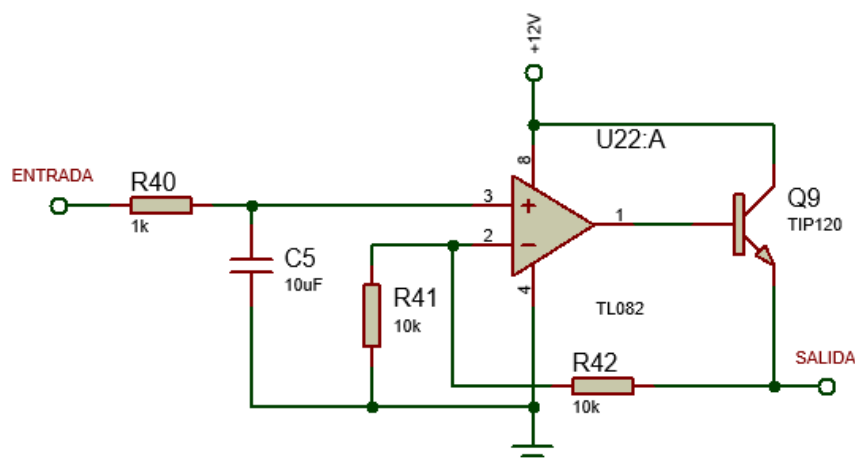


Figura 4.8. Circuito de salida analógica.

4.5.2. Circuito Integrado TL082

El circuito integrado TL082 tiene internamente 2 amplificadores operacionales de entrada de alta velocidad J-FET.

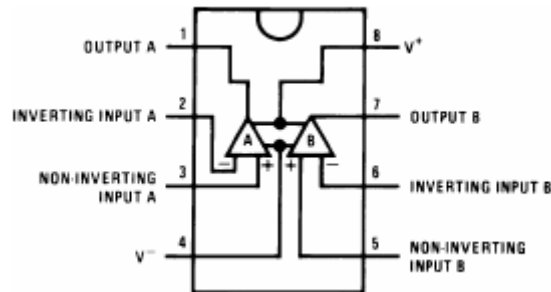


Figura 4.9. Circuito integrado TL082. En: <https://www.ti.com/lit/ds/symlink/tl082-n.pdf>.

Dentro de sus características tenemos:

- Amplia modo común (hasta $V_{CC} +$) y el rango diferencial.
- Polarización de entrada baja.
- Entrada alta impedancia.
- Compensación de frecuencia interna.
- Voltaje de alimentación de hasta: 18VDC.
- Voltaje de entrada de hasta 15V.
- Potencia total: 680 mW.

4.5.3. Circuito Integrado TIP120

El circuito integrado TIP120 es un transistor en configuración Darlington NPN con la finalidad de brindar ganancia en corriente. Para una mejor vista del circuito

mostramos la figura 4.10, donde se muestra la carcasa del circuito además de sus componentes internos.

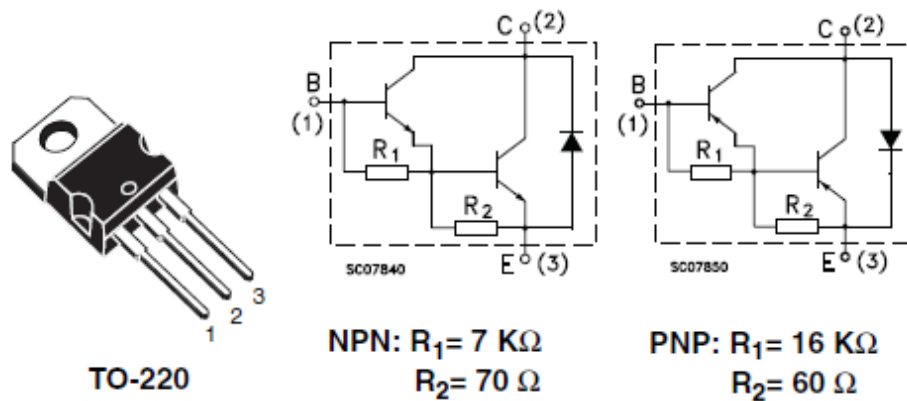


Figura 4.10. Circuito integrado TIP120, en:
https://www.electronicoscaldas.com/datasheet/TIP120-TIP121-TIP122-TIP125-TIP126-TIP127_ST.pdf

Dentro de las características más saltantes, se encuentran las siguientes:

- Dispositivo diseñado para operar como amplificador de propósito general y en conmutación de baja velocidad.
- IC max: 5A.
- IC pico max: 8A.
- IB max: 0.12A.
- Potencia total: 65W.
- VCEO: 60V, VCBO: 60V, VEBO: 5V.
- hFE min: 1000 (@ IC=3A, VCE=3V).
- Diodo damper entre colector y emisor.
- Bajo voltaje de saturación colector-emisor.
- Encapsulado: TO-220.

4.6. DISEÑO DE ENTRADA DIGITAL

Un requerimiento más de la tarjeta de adquisición es poseer 8 entradas digitales para leer voltajes de hasta 24 voltios. Como ya sabemos, los dispositivos digitales trabajan con niveles lógicos de 0 y 1, equivalentes a 0 voltios y 5 voltios respectivamente. Si colocamos directamente la entrada de 24 voltios a la etapa digital, ésta sufriría un gran daño, es por ello que se necesita de una etapa de acondicionamiento de señal.

4.6.1. Etapa de acondicionamiento de señal

Para obtener una entrada digital de 0 y 5 voltios, implementaremos una configuración basada en un optotransistor de baja potencia como es el circuito 4n25 que funcione como un interruptor, es decir, cuando se conmute el interruptor de la entrada, el LED del 4n25 se activará internamente y esto permitirá que se excite la base del transistor, haciendo que entre el colector y el emisor se cortocircuitará haciendo que en el colector del transistor se coloque un 0VDC, y que al pasar por el inversor se convertirá en un 5VDC que será un 1 lógico en la entrada del microcontrolador.

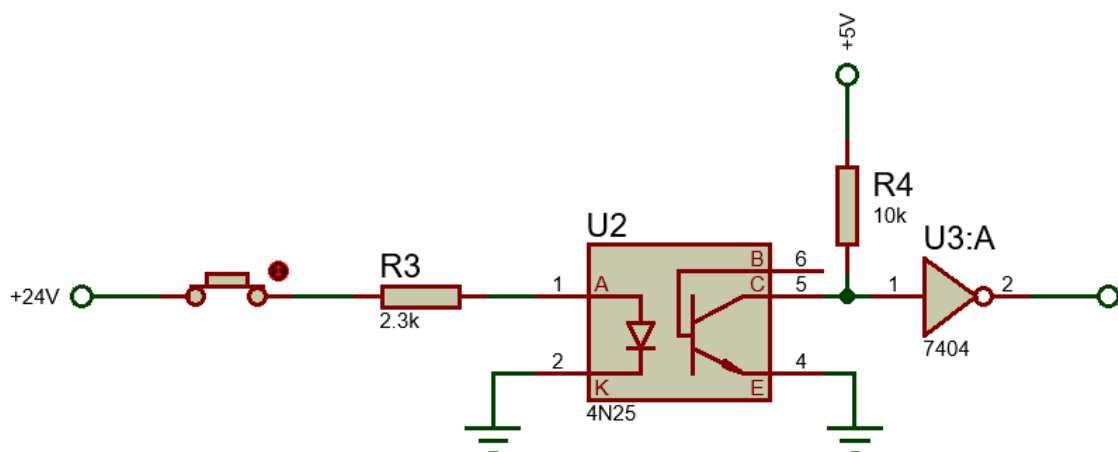


Figura 4.11: Etapa de acondicionamiento de señal digital.

4.6.2. Circuito integrado 4n25

El circuito integrado 4n25 es un tipo de optoacoplador, denominado opto transistor que al ingresar una corriente directa al LED integrado en el circuito, este excitará la base del transistor a través de la luz, y hará que circule corriente entre el colector y emisor.

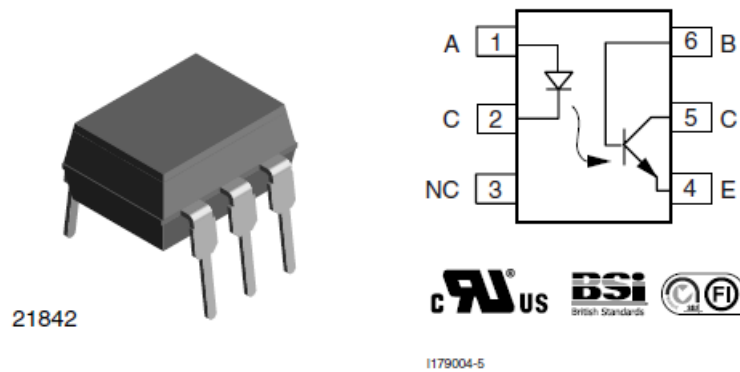


Figura 4.12: Circuito integrado 4n25. En:
<https://www.vishay.com/docs/83725/4n25.pdf>.

A continuación detallamos algunas características técnicas del circuito:

Entrada

- Voltaje inverso: 5V
- Corriente directa: 60mA
- Pico corriente: $3A < 10\mu s$
- Disipación de potencia: 100mW

Salida

- Voltaje colector-emisor: 70V

- Voltaje base-emisor: 7V
- Corriente colector: 50mA
- Disipación de potencia: 150mW

4.7. DISEÑO DE SALIDA DIGITAL

Un requerimiento más de la etapa de adquisición de datos es la de manejar 8 salidas digitales de por lo menos 24 voltios, para esto utilizaremos, desde luego una etapa de potencia para conseguir justamente los 24 voltios o más, debido a que la etapa digital generará sólo una salida máxima de 5 voltios.

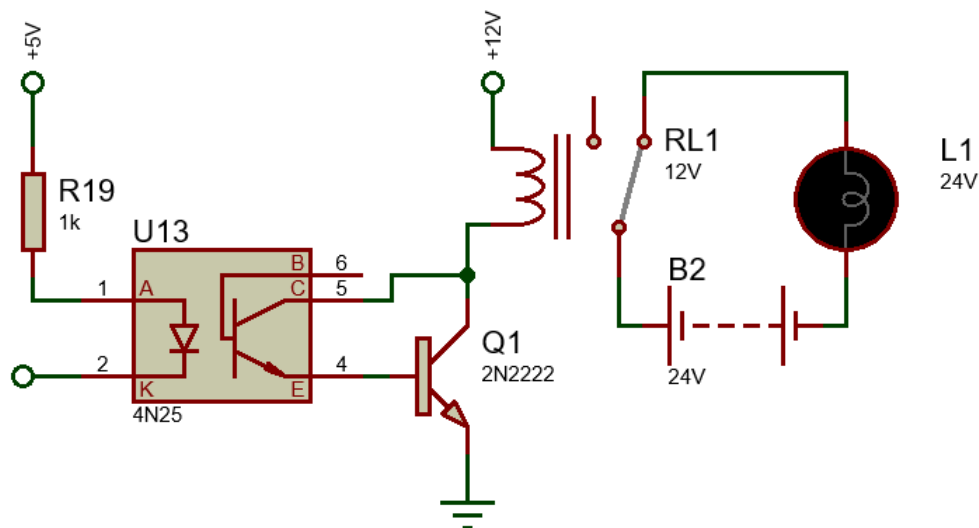


Figura 4.13: Transmisor de Temperatura modelo YTA de YOKOGAWA

De acuerdo a lo visto en el circuito de la figura 4.13, se utilizará un optoacoplador tipo transistor el cual se activará al enviar una señal de 0 voltios al cátodo del diodo interno del dispositivo, con ello el diodo se encenderá y con ello el optotransistor se activará haciendo circular una corriente en colector y el emisor el cual está conectado a un transistor externo el 2N2222, haciendo una configuración Darlington, esto hace que circule nuevamente una corriente entre el colector y el emisor del transistor 2N2222, haciendo que a su vez circule la corriente por la bobina del relevador.

Al circular la corriente por el relé hará que funcione como un electroimán atrayendo el contacto y cerrando un lado del contacto, abriendo el otro.

4.7.1. Relevador

Un relevador o simplemente relé o relay (en inglés), es un interruptor cuyo control corre por cuenta de un circuito eléctrico. a través de una bobina y un electroimán incide sobre diversos contactos para la apertura o el cierre de otros circuitos, que funcionan de manera independiente. Lo que hace la bobina al incidir una corriente sobre ella es generar un campo magnético que lleva los contactos a establecer una determinada conexión tomando en cuenta los contactos existentes. El electroimán, por su parte, permite el cierre de los contactos. De esta forma, el relevador actúa como un interruptor que puede fomentar el paso de la corriente eléctrica por dichos contactos o su interrupción.

Los relevadores, en definitiva, permiten desarrollar una conmutación a distancia, controlando altas tensiones y corrientes a través de sus contactos, con un bajo voltaje y corriente que pasen por su bobina.

Para nuestro caso se utilizará un relevador cuya bobina sea de 12 voltios continuos y de 3 contactos.

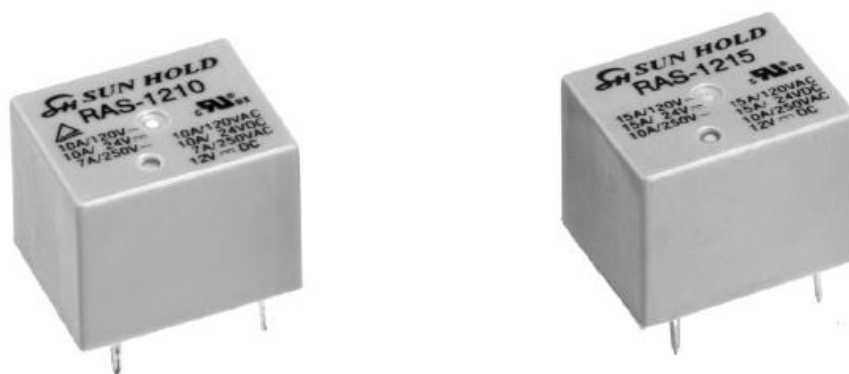


Figura 4.14: Relevador modelo RAS-1210 de la marca SUN HOLD, en:

<http://www.sunhold.com/upload/prd1/118-3.pdf>

Dentro de sus características más saltantes, tenemos:

- Voltaje nominal de 12 voltios continuos.

- Resistencia de la bobina de $400\pm 40\Omega$ a 20°C .
- Voltaje de operación mínimo de 9 voltios continuos.
- Consumo de potencia de 0.36W a 20°C
- Rango de temperatura de operación de -30°C a $+80^{\circ}\text{C}$.
- Arreglo de contactos tipo 1 en forma C.
- Material de los contactos: aleación de plata.
- Capacidad del contacto: 10 amperios a 120 voltios alternos o 24 voltios continuos, resiste 7 amperios a 250 voltios alternos.
- Resistencia del contacto de $50\text{m}\Omega$.
- Vida eléctrica: 20 000 operaciones.
- Vida mecánica: 10 000 000 de operaciones (1800 operaciones por hora).

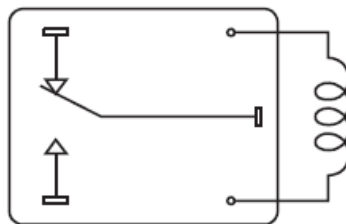


Figura 4.15: Esquema interno del relevador seleccionado, dibujo en:

<http://www.sunhold.com/upload/prd1/118-3.pdf>

4.8. SOFTWARE DE PROGRAMACIÓN EN LENGUAJE ESCALERA

El software de programación en lenguaje escalera que se utiliza para programar Arduinos es el Waltech Lader Maker, el cual es un desarrollo de programación abierta, el cual permite la programación en lenguaje escalera de 3 tipos de Arduino, entre los cuales se encuentra el Arduino MEGA, el Arduino NANO y el Arduino 1.

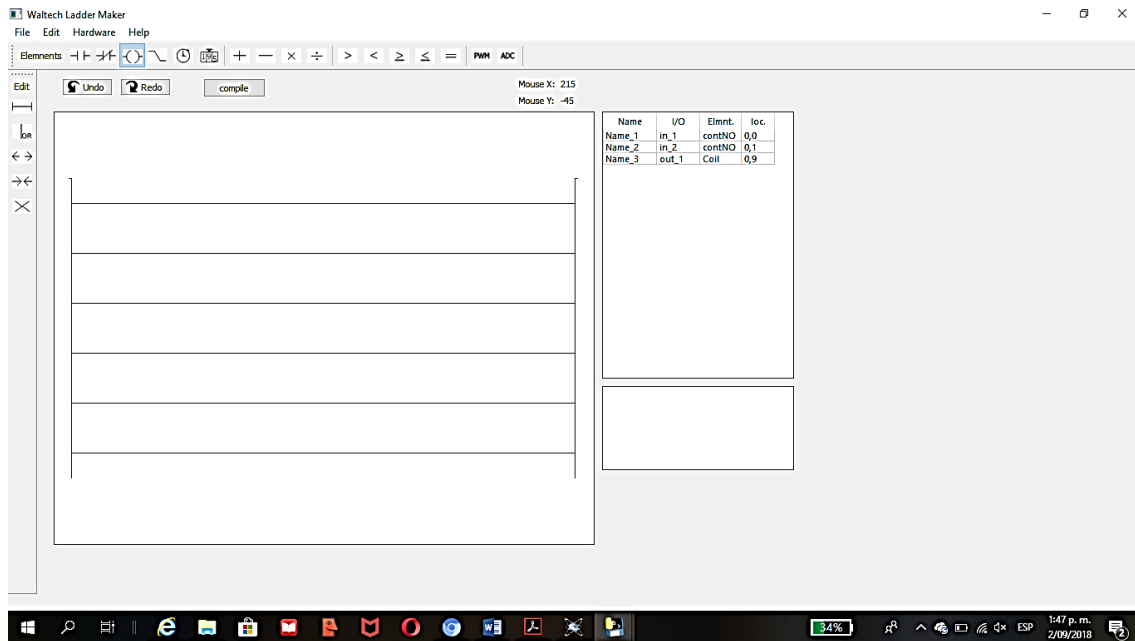


Figura 4.16: Pantalla del software de programación Waltech Lader Maker.

De acuerdo a la cantidad de entradas y salidas seleccionadas, se ha optado por el Arduino MEGA, por ello se va a tener que tomar en cuenta los pines que previamente tiene configurados en software de programación, cuyo esquema se presenta en la figura 4.17.

Como se puede apreciar en el esquema de la figura 4.17 y tomando en cuenta lo que requiere nuestro proyecto, tomaremos en cuenta las entradas y salidas de lo que se considera en el software de programación con respecto a los pines que utiliza el Arduino MEGA.

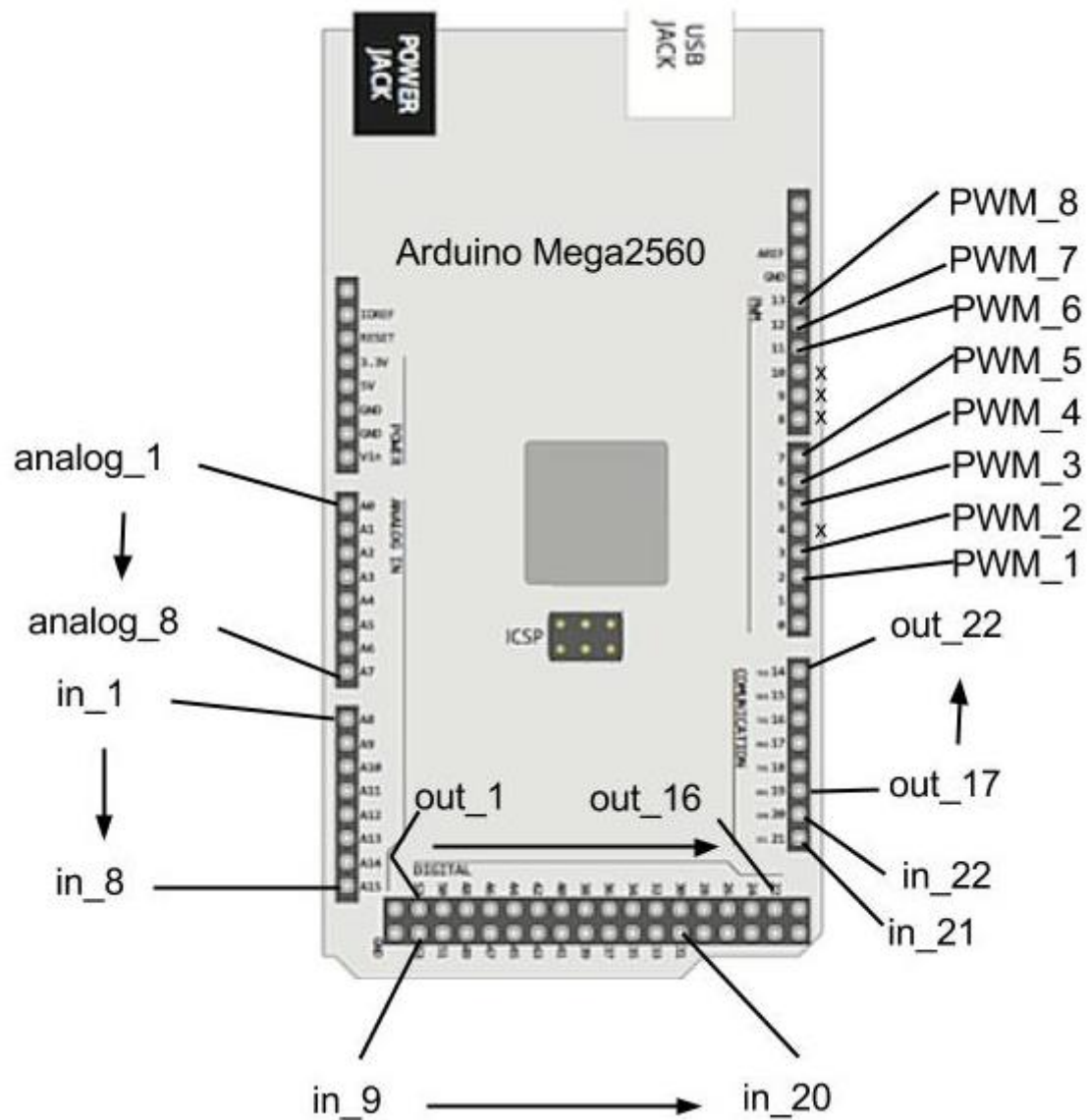


Figura 4.17: Esquema de Arduino MEGA donde se indica el esquema de entradas y salidas que maneja el software de programación Waltech Lader Maker correspondiente al Arduino MEGA.

Descripción de entradas y salidas	Puertos en Software	Puertos del Arduino MEGA
Entradas digitales	in_1	A8
	in_2	A9
	in_3	A10
	in_4	A11
	in_5	A12
	in_6	A13
	in_7	A14
	in_8	A15
Salidas digitales	out_1	P52
	out_2	P50
	out_3	P48
	out_4	P46
	out_5	P44
	out_6	P42
	out_7	P40
	out_8	P38
Entradas analógicas	analog_1	A0
	analog_2	A1
	analog_3	A2
	analog_4	A3
Salidas analógicas	PWM_1	P2
	PWM_2	P3
	PWM_3	P5
	PWM_4	P5

Tabla 4.2. *Tabla de entradas y salidas del software con respecto a las entradas y salidas del Arduino MEGA.*

4.9. DISEÑO DEL HARDWARE DEL PLC

A continuación se mostrarán todos los circuitos que son parte del PLC diseñado, basado en el Arduino MEGA.

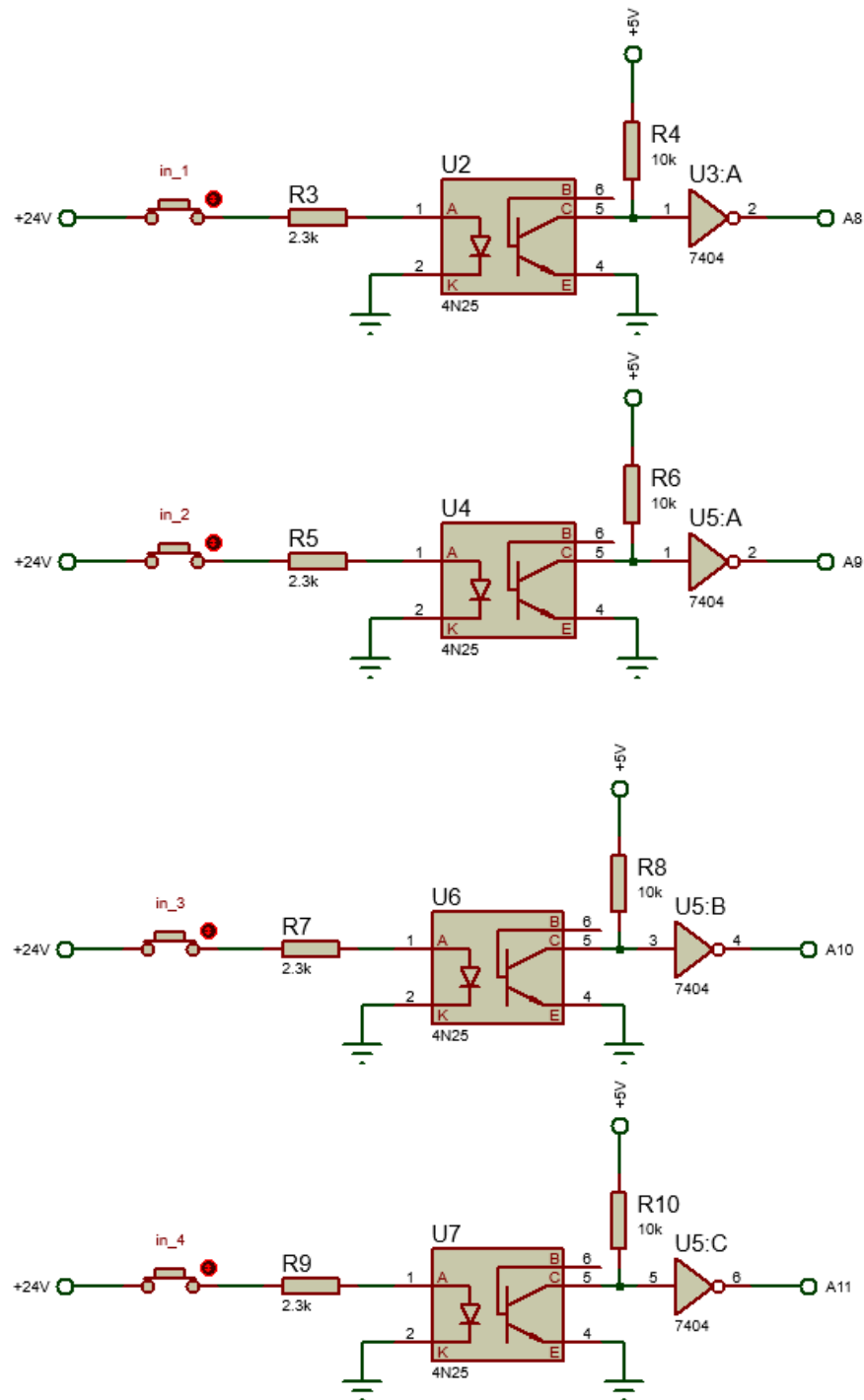


Figura 4.18: Esquema de las primeras 4 entradas digitales: in_1, in_2, in_3, in_4.

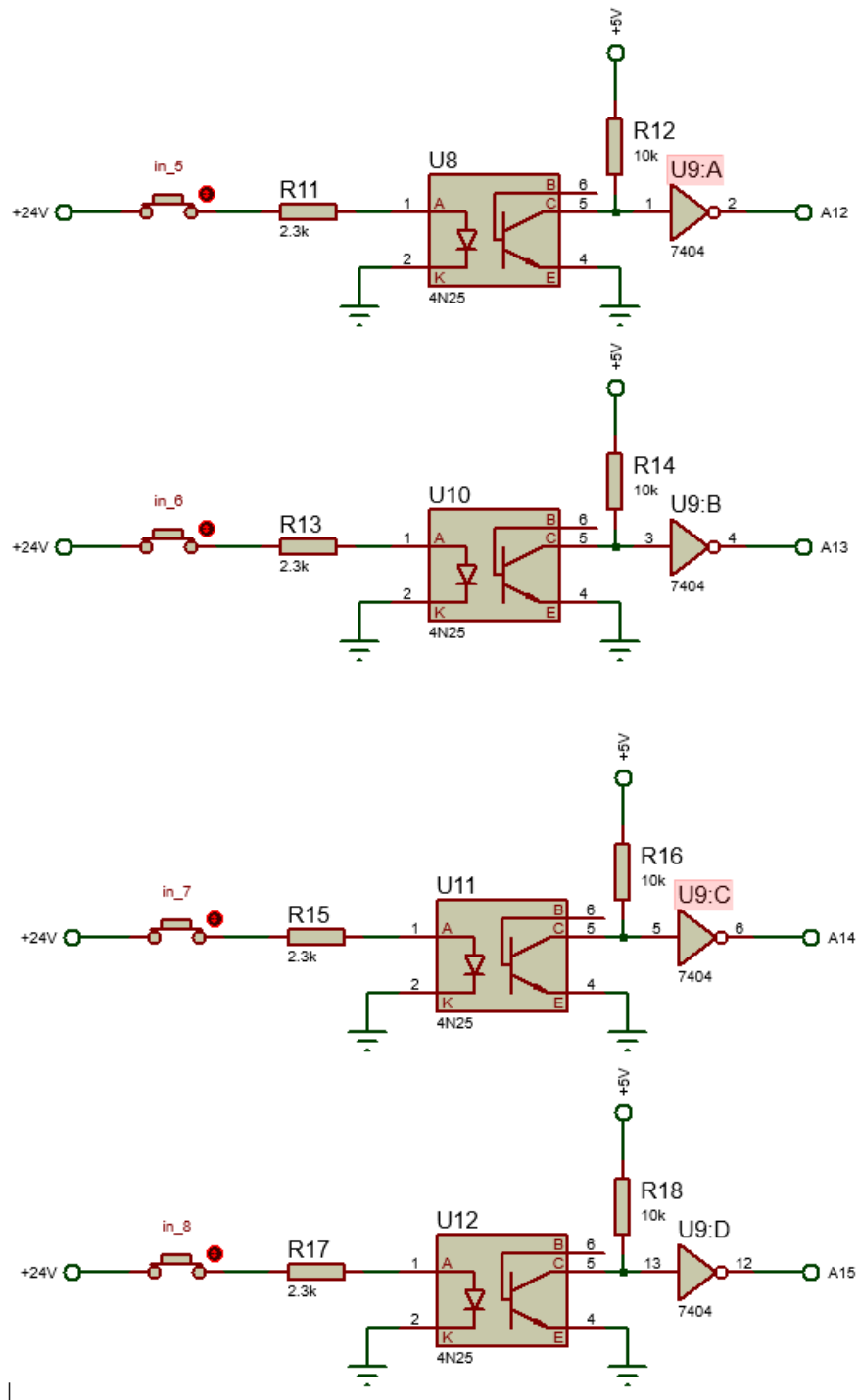


Figura 4.19: Esquema de las siguientes 4 entradas digitales: in_5, in_6, in_7, in_8.

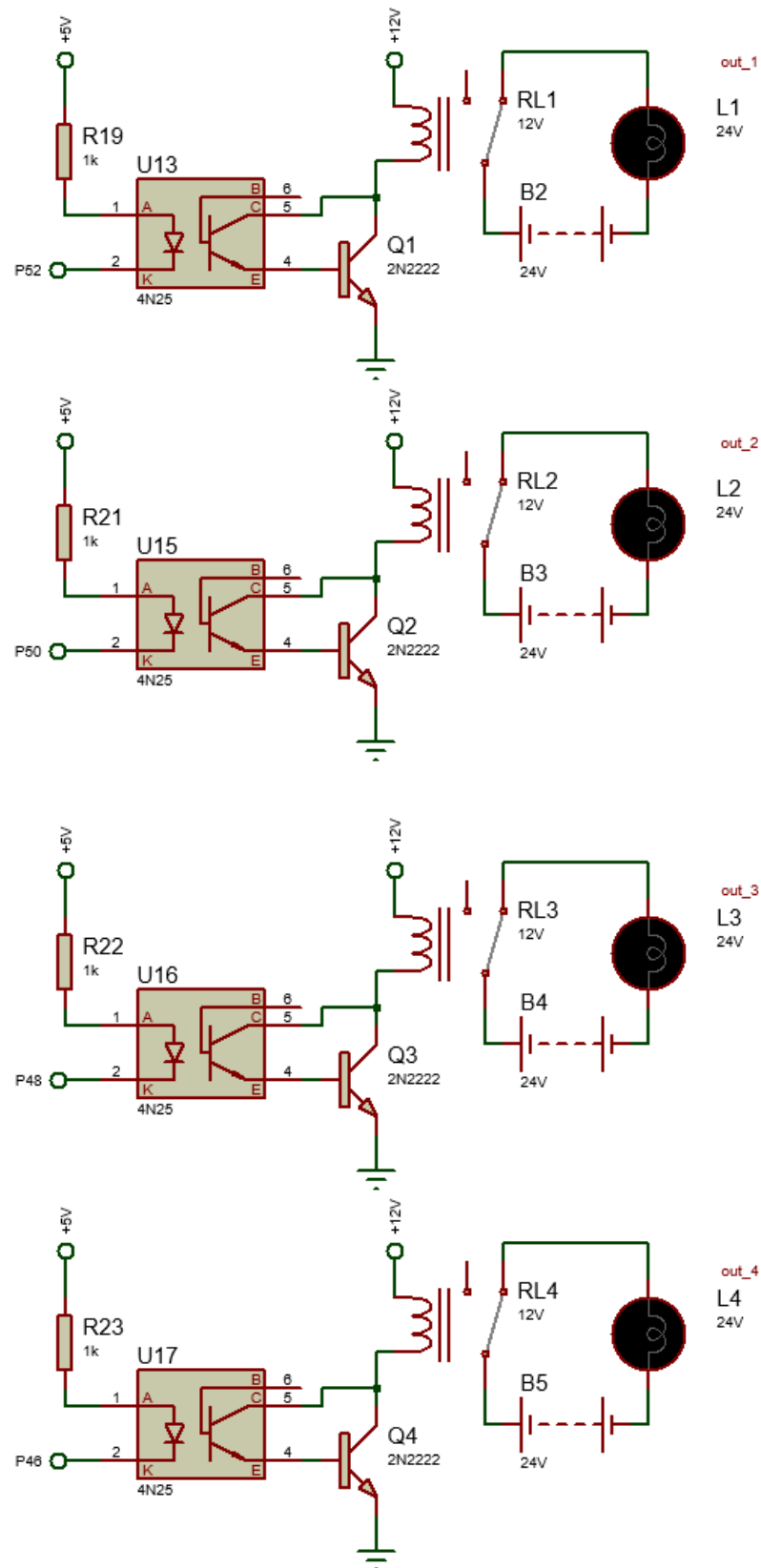


Figura 4.20: Esquema de las salidas digitales: out_1, out_2, out_3, out_4.

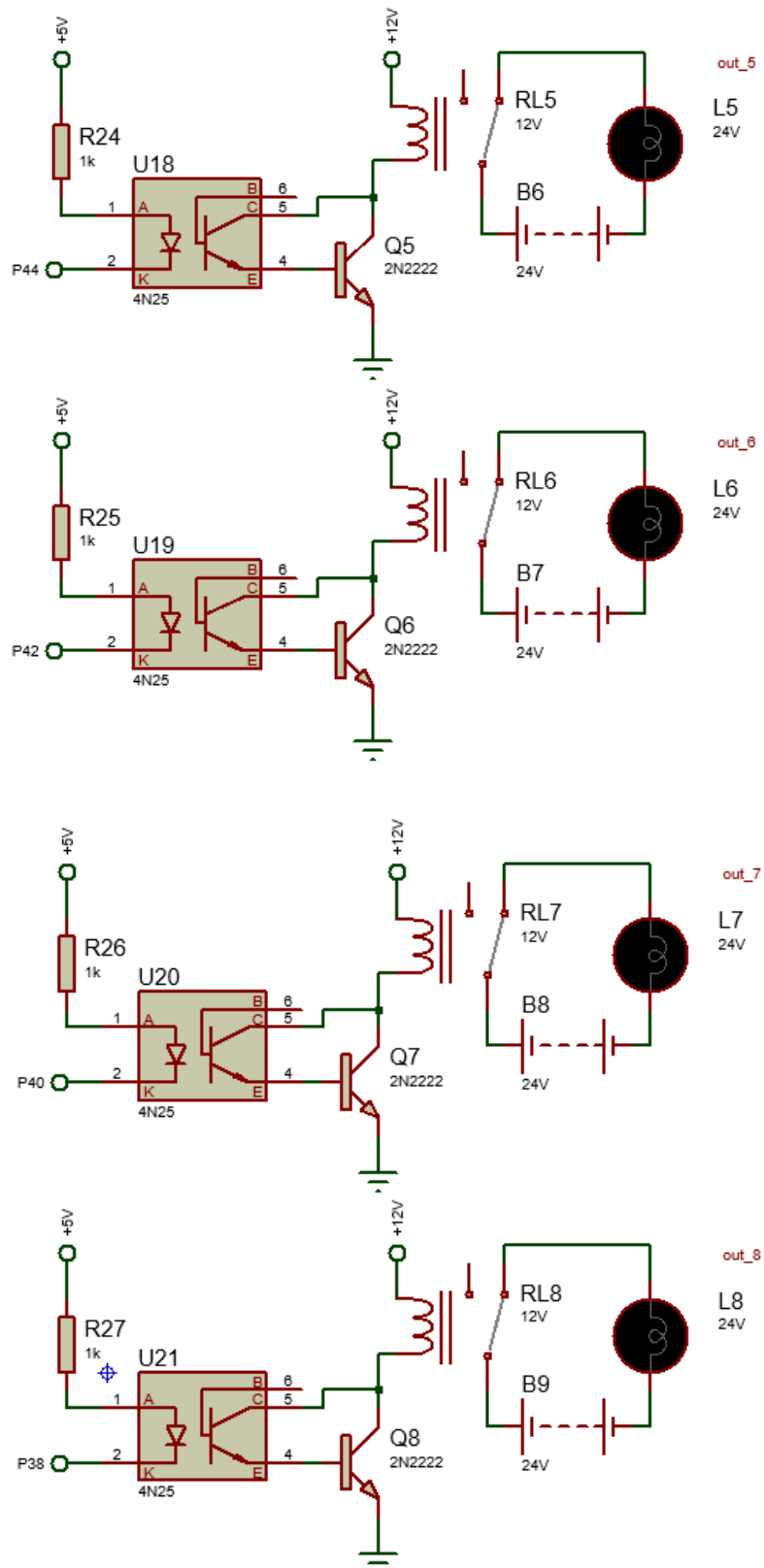


Figura 4.21: Esquema de las salidas digitales: out_5, out_6, out_7, out_8.

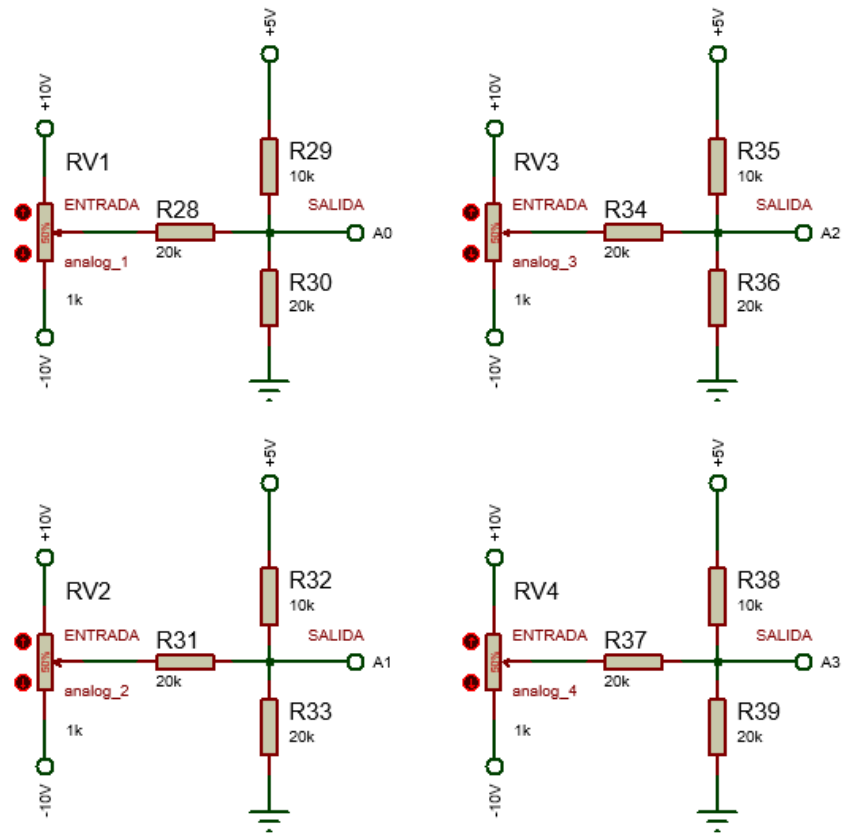


Figura 4.22: Esquema de las entradas analógicas: analog_1, analog_2, analog_3, analog_4.

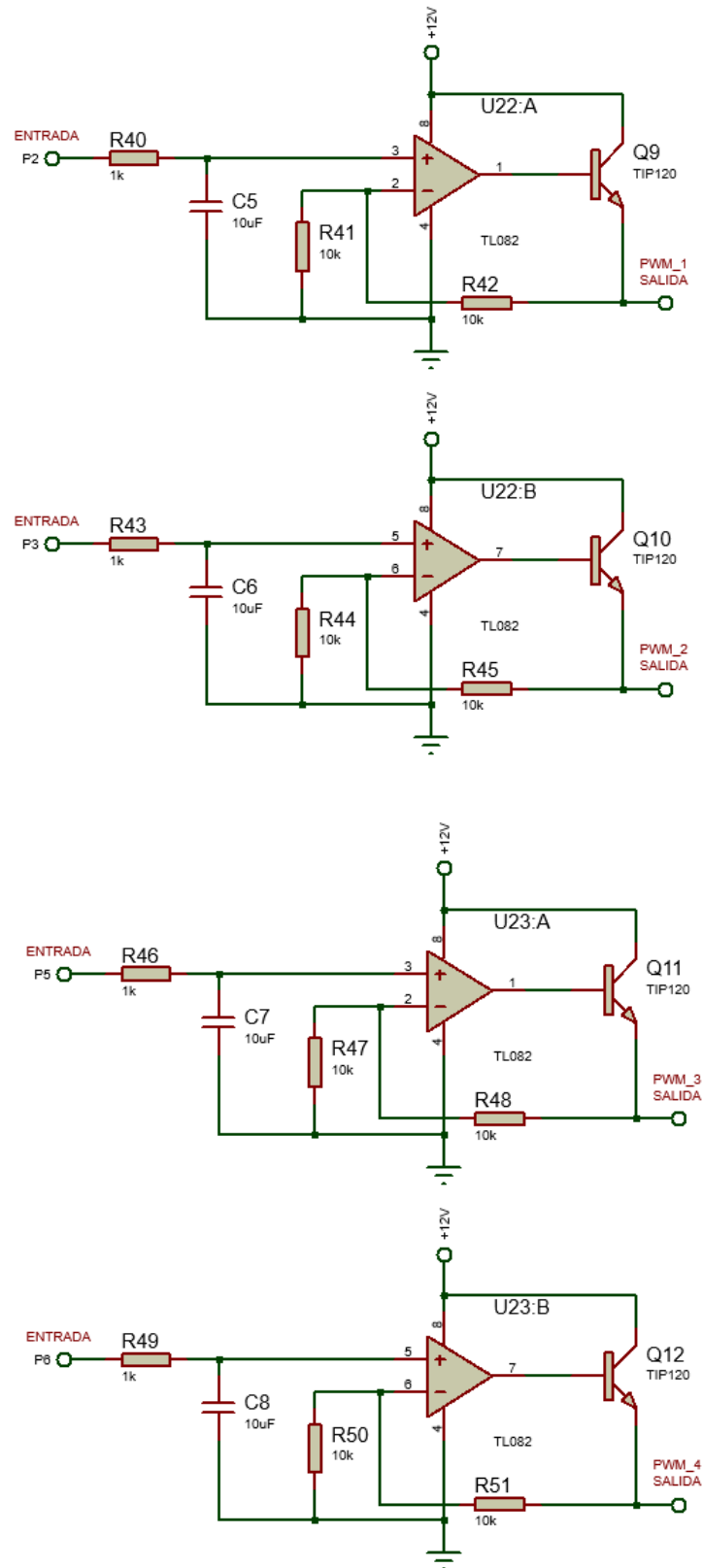


Figura 4.23: Esquema de las salidas analógicas: PWM_1, PWM_2, PWM_3, PWM_4.

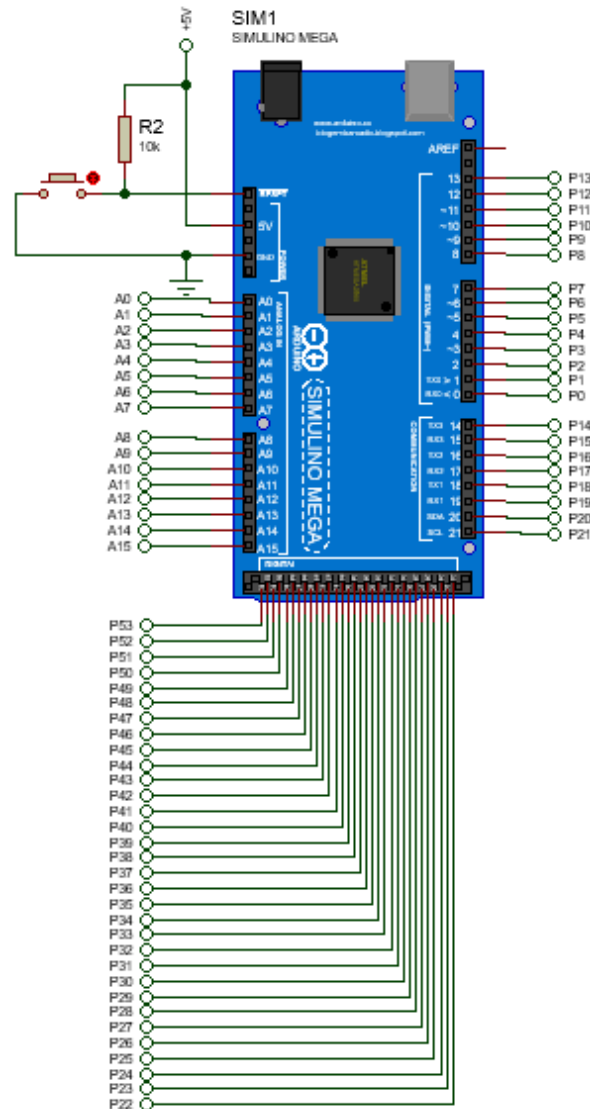


Figura 4.24: Esquema del Arduino MEGA en Proteus.

4.10. PRUEBA DE SOFTWARE DE PROGRAMACIÓN EN LENGUAJE ESCALERA

A continuación se va a realizar un ejemplo para la realización de un cambio de giro de un motor de corriente continua.

Para ello se va a utilizar 3 entradas digitales, las cuales detallamos a continuación:

- in_1: Horario
- in_2: Anti horario

- in_3: Detener

Además, se utilizarían 2 salidas, las cuales detallamos:

- Out_1: Salida 1
- Out_2: Salida 2

El esquema quedaría tal cual se muestra en la figura 4.25, al dar compilar se carga el archivo directamente al controlador.

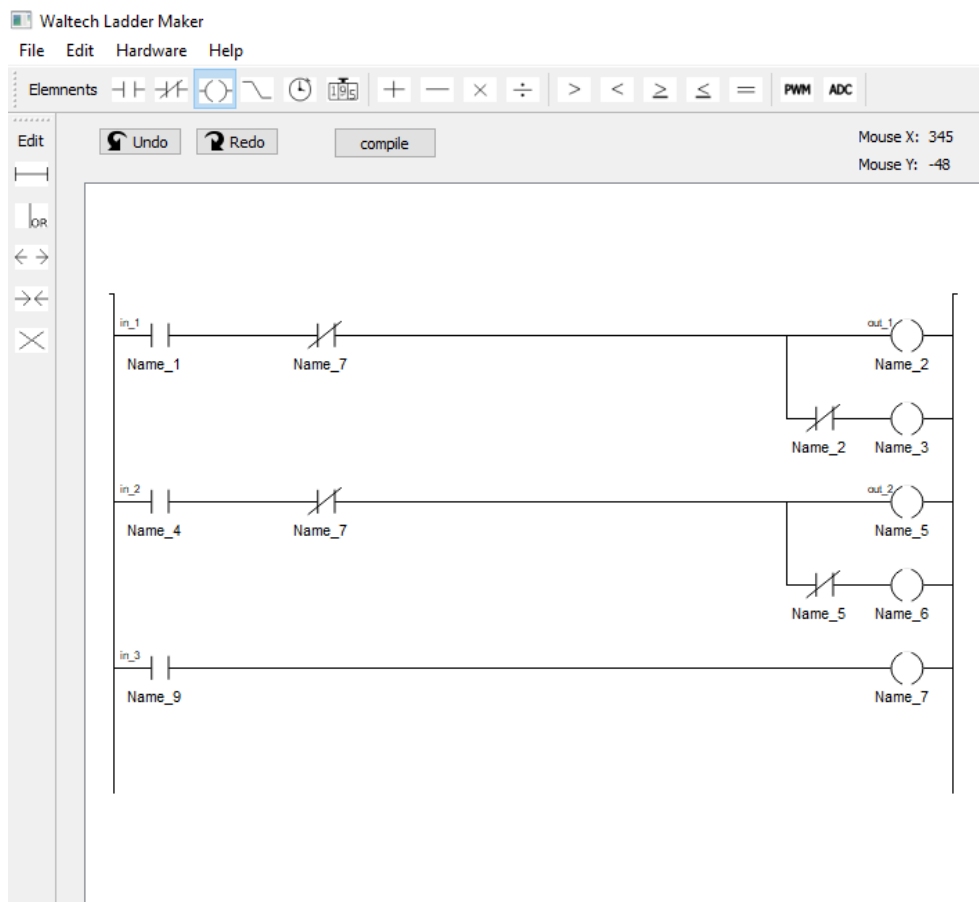


Figura 4.25: Esquema de programación en lenguaje escalera.

CAPÍTULO 5: CONCLUSIONES

- Se diseñó un programador lógico programable usando microcontrolador ATMEGA y se programó el PLC utilizando software con la opción de trabajar con lenguaje LADDER para aplicaciones de laboratorio.
- Se diseñó la placa de desarrollo basada en el microcontrolador ATMEGA para la aplicación del controlador lógico programable, considerando cada una de sus partes.
- Se diseñó los circuitos de acondicionamiento de señal para las señales de entrada analógica y digital, considerando arreglos de resistencias para la entrada analógica de -10V a +10V y opto acopladores con opto transistor para las entradas digitales, tal como se muestra en el capítulo 4 de la presente tesis.
- Se diseñó los circuitos de etapa de potencia para las señales de salida analógica y digital, considerando amplificadores de voltaje y de corriente para la salida analógica, y salida tipo relé para la salida digital, tal como se muestra en el capítulo 4 de la presente tesis.
- Se seleccionó cada uno de los dispositivos necesarios para el correcto funcionamiento del controlador lógico programable.
- Se realizó las pruebas a nivel de hardware y software del controlador lógico programable, de la cual tenemos descrita la de cambio de giro de un motor de corriente continua, como se muestra en el capítulo 4 de la presente tesis.

CAPÍTULO 6: BIBLIOGRAFÍA

- Arduino, **Arduino**. En: <https://www.arduino.cc/>, 1 pp.
- Arduino, **Arduino Uno**. En: <https://store.arduino.cc/usa/arduino-uno-rev3>, 1 pp.
- Arduino, **Arduino Leonardo**. En: <https://store.arduino.cc/usa/arduino-leonardo-with-headers>, 1 pp.
- Arduino, **Arduino 101**. En: <https://store.arduino.cc/usa/arduino-101>, 1 pp.
- Arduino, **Arduino Esplora**. En: <https://store.arduino.cc/usa/arduino-esplora>, 1 pp.
- Arduino, **Arduino Micro**. En: <https://store.arduino.cc/usa/arduino-micro>, 1 pp.
- Arduino, **Arduino Nano**. En: <https://store.arduino.cc/usa/arduino-nano>, 1 pp.
- Arduino, **Arduino Mega**. En: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>, 1 pp.
- Arduino, **Placa de expansión Arduino Motor Shield REV3**. En: <https://store.arduino.cc/usa/arduino-motor-shield-rev3>, 1 pp.
- Arduino, **Placa de Expansión Arduino Gsm Shield 2 (Integrated Antenna)**. En: <https://store.arduino.cc/usa/arduino-gsm-shield-2-integrated-antenna>, 1 pp.
- Arduino, **Placa de Expansión Arduino Usb Host Shield**. En: <https://store.arduino.cc/usa/arduino-usb-host-shield>, 1 pp.
- Arduino, **Placa de Expansión Arduino Wifi Shield**. En: <https://store.arduino.cc/usa/arduino-wifi-shield>, 1 pp.
- Arduino, **Kit Arduino Starter Kit Multi-Language**. En: <https://store.arduino.cc/usa/arduino-starter-kit>, 1 pp.
- Arduino, **Pantalla de cristal líquido para Arduino**. En: <https://store.arduino.cc/usa/arduino-lcd-screen>, 1 pp.
- Arduino, **Impresora 3D Arduino Materia 101 - Assembled**. En: <https://store.arduino.cc/usa/arduino-starter-kit>, 1 pp.
- National Semiconductor. **LM78XX Series Voltage Regulators**. En: http://pdf.datasheetcatalog.com/datasheets/150/44435_DS.pdf, 2000, 10 pp.
- ST. **TIP120, TIP121, TIP122, TIP125, TIP126, TIP127 Complementary power Darlington transistors**. En: https://www.electronicoscaldas.com/datasheet/TIP120-TIP121-TIP122-TIP125-TIP126-TIP127_ST.pdf, 2008, 13 pp.
- SUN HOLD. RAS Series. En: <http://www.sunhold.com/upload/prd1/118-3.pdf>, 4 pp.
- Texas Instruments. **TL082 Wide Bandwidth Dual JFET Input Operational Amplifier**. En: <https://www.ti.com/lit/ds/symlink/tl082-n.pdf>, 2013, 23 pp.

- VISHAY. **4N25, 4N26, 4N27, 4N28 Optocoupler, Phototransistor Output, with Base Connection.** En: <https://www.vishay.com/docs/83725/4n25.pdf> , 2017, 7 pp.
- Waltech. Open Source Designs. En: <http://www.waltech.com/open-source-designs/>, 2018, 1 pp.
- Wikipedia, **Controlador Lógico Programable.** En: https://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable, 2018, 1pp.
- Wikipedia, **Lenguaje LADDER.** En: https://es.wikipedia.org/wiki/Lenguaje_Ladder, 2018, 1 pp.